

## Testbed to Verify Interoperability among Heterogeneous Devices with OPC UA

Byunghun Song<sup>1</sup>, Hojoong Kim<sup>2</sup>, Wonhee Lee<sup>3</sup> and Junho Shin<sup>\*</sup>

*Korea Electronics Technology Institute (KETI)*

<sup>1</sup>bhsong@keti.er.kr, <sup>2</sup>khj123abc@keti.re.kr, <sup>3</sup>zsed50@keti.re.kr, <sup>\*</sup>jhshin@keti.er.kr

### Abstract

*Recently, in the manufacturing industry, many people have tried to lower production cost and secure a flexible production system for customers' demands using the Industrial Internet of Things. One of core components to establish the Industrial Internet of Things is interoperability among heterogeneous devices, and OPC UA (OPC Unified Architecture) is receiving much attention as a communication standard to realize it. However, it is hard for customers to select a proper solution because OPC UA has been developed by various companies and organizations, with varying functions and levels of realization. This study conducted a comparative analysis on major commercial solutions and open sources used to realize OPC UA, and identified supported core functions and level of their realization. Also, this study established a testbed to realize an OPC UA devices using different commercial solutions and open sources, and interlocked industrial devices and environment sensor, etc. to verify interoperability among heterogeneous devices with OPC UA.*

**Keywords:** OPC UA, Industrial Internet of Things, interoperability, testbed

### 1. Introduction

Recently, to solve problems in the manufacturing industry including intensifying competition and aging work force, the Germany took the lead in making IEC SMB SG8 and established RAMI 4.0 (Reference Architectural Model Industry 4.0)[1]. And the USA announced IIRA (Industrial Internet Reference Architecture)[2] via IIC to innovate manufacturing based on the convergence of manufacturing and ICT. For this trend, global companies have tried to lower production cost and secure a flexible production system for customers' demands using the Industrial Internet of Things technology, by applying ICT-based smart manufacturing technologies to the production system.

One of core components to establish the Industrial Internet of Things is interoperability among heterogeneous devices. So far, various industrial communication protocols have been developed to realize interoperability among heterogeneous devices. But in reality, it is hard to realize interoperability among various heterogeneous devices due to technological dependency to vendors and technical closeness. In this situation, OPC UA[3] is receiving much attention as a communication standard to realize interoperability among various heterogeneous devices.

OPC UA is a data exchange standard standardized as IEC62541 by IEC TC57 group in 2012, to guarantee the safety and reliability of industrial communication. OPC UA is not dependent on programming language and OS to run the application. Therefore, it can be operated regardless of manufacturer and platform, as well as can be utilized in the most fields of manufacturing industry[4].

Though many companies and organizations have actively developed various commercial solutions and open sources for OPC UA, it is hard for customers to select a

---

Received (July 2, 2017), Review Result (October 1, 2017), Accepted (October 16, 2017)

proper solution for development because they have varying functions and levels of realization. In addition, there are not enough testbeds to test interoperability by interlinking the OPC UA product with various solutions.

This study conducted a comparative analysis on commercial solutions and open sources, developed by various companies and organizations to realize OPC UA. Also, this study established a testbed to realize OPC UA devices using different commercial solutions and open sources, and interlinked industrial devices and environment sensor, *etc.* to verify interoperability among heterogeneous devices with OPC UA.

This article is composed as follows. Chapter 2 explains about OPC UA. Chapter 3 explains about a comparative analysis on the commercial solutions and open sources used to realize OPC UA. Chapter 4 explains about the composition of OPC UA server and OPC UA client using commercial solutions and open sources, as well as the establishment of testbed. And Chapter 5 summarizes the contents of this study and describes the future task.

## **2. OPC UA**

OPC UA is an independent industrial communication protocol for manufacturing companies, based on the server-client principle. OPC UA server is connected to industrial devices to collect and manage relevant information. OPC UA client accesses to OPC UA server to request to collect and manage relevant information.

OPC UA defines information model of the device as node. The defined node is integrated as a hierarchical structure in the address space of OPC UA[5][6]. OPC UA supports communication of the information defined as a hierarchical structure between OPC UA server and OPC UA client, using the defined core functions.

Core functions of OPC UA are Data Access, Methods, Events, Alarms & Conditions and Historical Access. Data Access is a function to check data delivered from OPC UA server via OPU UA client in real time. Methods is a function to call a specific function of OPC UA server to make OPC UA server conduct it via OPC UA client. Events is a function to notify an important event occurred in the system with OPC UA server to OPC UA client. Alarms & Conditions is a function to notify an important event, occurred in the system with OPC UA server and should be responded, to OPC UA, and then OPC UA client notifies its response to OPC UA server. Historical Access is a function to save data and event information, as well as to inquire history.

OPC Foundation[3] has standardized OPC UA and developed related technologies. In particular, OPC Foundation has maintained and released 3 OPC UA stacks composed of C, Java and .Net, respectively. And many other companies and organizations have developed and released solutions to realize their own OPC UA. Based on the cooperation with various organizations including PLCopen[7], AIM[8], BACnet[9] and FDI[10], OPC UA has been developed as a way to deliver a unique information model.

## **3. Solutions to realize OPC UA**

Solutions to realize OPC UA have been developed by various companies and organizations. They are divided into commercial solutions and open sources. Table 1. shows the commercial solutions and open sources to realize OPC UA analyzed in this study.

### **3.1. Commercial Solutions**

Commercial solutions to realize OPC UA have been developed by each company's own technologies. And the number of companies developing OPC UA is increasing rapidly. Most commercial solutions are provided as both paid license version and free trial version. Unified Automation's OPC UA SDK is a commercial SDK solution to develop

OPC UA server and OPC UA client[11]. Starting with the C++ based SDK released on April 2009, it supported various programming languages including C, C++, .NET and Java, and all core functions are already established. It provides the free trial version in which user can use OPC UA SDK without purchasing a license, but the trail version supports only few platforms and expires after a period of time. To establish a limitless OPC UA, user must purchase a Binary license or Source Code license.

Softing's OPC UA Development ToolKits is a commercial SDK solution to develop OPC UA server and OPC UA client[12]. It supports programing languages such as C, C++ and .NET, and all core functions are already established. OPC UA Development ToolKits also provides the free trial version, in which user can use OPC UA SDK without purchasing a license, but the hours of use and available platforms are limited. Therefore, to realize a limitless OPC UA function, user must purchase a Binary license or Source Code license.

**Table 1. Commercial Solutions and Open Sources to Realize OPC UA**

Name	Commercial Solution/Open Source	License	Programming Language	Server and Client Support	Established Core Functions	Remark
Unified Automation OPC UA SDK	Commercial solution	Binary, Source Code	C, C++, Java, .NET	All	Data Access, Methods, Events, Alarm & Conditions, Historical Access	Provides the Free Trial Version
Softing OPC UA Development Toolkits	Commercial Solution	Binary, Source Code	C, C++, .NET	All	Data Access, Methods, Events, Alarm & Conditions, Historical Access	Provides the Free Trial Version
KEPServerEX	Commercial Solution	Subscription, Perpetual	.NET	All	Data Access	Provides the Free Trial Version
Modigm Server Modigm Client Tool Kit	Commercial Solution	Binary	.NET	All	Data Access	
Prosys Opc Ua	Commercial Solution	Binary, Source Code	C, C++, Java, .NET	All	Data Access, Methods, Events, Alarm & Conditions, Historical Access	Provides the Free Trial Version
open62541	Open Source	MPL 2.0	C, C++	All	Data Access, Methods	
node-opcua	Open Source	MIT	Javascript	All	Data Access, Methods, Events	
FreeOpcUa	Open Source	LGPL 3.0	C++, Python	All	Data Access, Methods, Events, Historical Access	

Prosys OPC's Prosys OPC UA is a commercial SDK solution to develop OPC UA server and OPC UA client[13]. It supports various programming languages such as C, C++, Java and .NET, and all core functions are already established. Like Unified Automation and Softing, Prosys OPC provides the free trial version, but the hours of use

and available platforms are limited. To realize a limitless OPC UA, user must purchase a Binary license or Source Code license.

KEPSEServerEx is an industrial communication linkage platform developed by Kepware[14]. Since the Version 5.2.133.0 released on February 2010, KEPSEServerEx has supported OPC UA interface. It also provides various industrial protocols and drivers, so user can configure an industrial device with OPC UA server and OPC UA client easily through setting. Among core functions, KEPSEServerEx supports only Data Access. KEPSEServerEx provides Subscription license and Perpetual license.

Modigm's Modigm Server and Modigm Client Toolkit is an OPC UA solution developed by Modigm[15]. With Modigm Server, user can configure various devices with OPC UA server by adding plug-in type device drivers and adjusting a simple configuration setting. Among core functions, Modigm Server and Modigm Client Toolkit support only Data Access. In 2013, they were tested and certificated officially by OPC Foundation for the first time in Korea.

### 3.2. Open Sources

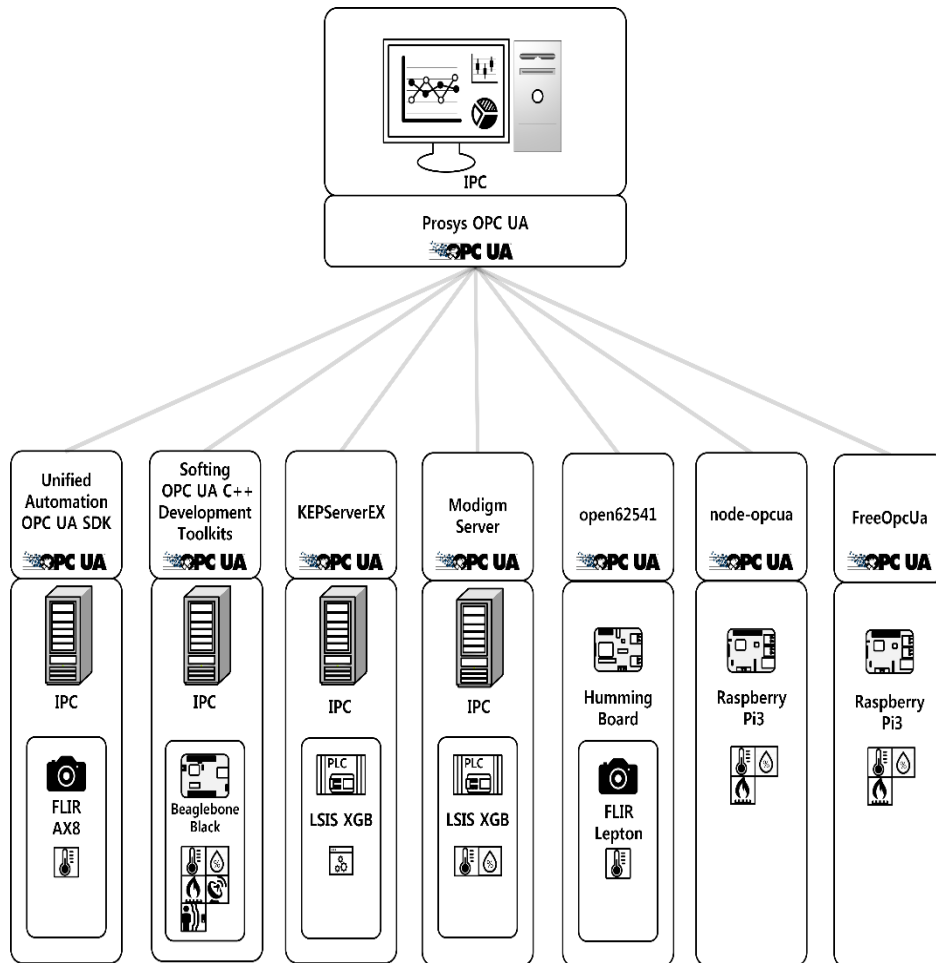
Open sources to realize OPC UA have been developed by nonprofit organizations. They have released codes developed by reflecting opinions of various developers. open62541 is an OPC UA open sources project of Franunhofer released since December 2013[16][17]. It supports C and C++, and currently version 0.2 was released officially. The source code is released via GitHub[18] with MPL v2.0 license[19]. Developer can realize OPC UA server and OPC UA client using the provided source code. However, open62541 version 0.2 does not support Events, Alarms & Conditions and Historical Access.

FreeOpcUa is an OPC UA open source project released by SINTEF since May 2014[20]. The source code is released with LGPL 3.0 license[21], and it supports Python and C++. Realization level of FreeOpcUa is different depending on programming language. Library developed with Python realized more functions than C++ version. Among the currently released OPC UA open sources, it provides the most functions, but doesn't provide Alarms & Conditions function

node-opcua is an OPC UA open source released since February 2014[22]. The source code is released via GitHub with MIT license. With it, user can realize OPC UA server and client. node-opcua was developed based on Javascript, and version 0.0.64 is released. It doesn't provide Historical Access function.

## 4. Testbed to Verify Interoperability

This study developed a testbed to check the realization status of currently developed OPC UA commercial solutions and open sources, and verify interoperability among heterogeneous devices on different platforms. Figure 1. shows the overall structure of the testbed to verify interoperability. The testbed is composed of 7 OPC UA servers and 1 OPC UA client. OPC UA server is realized with 4 commercial solutions and 3 open sources. OPC UA client is realized with 1 commercial solution. OPC UA server is connected to PLC, thermal camera, environment sensor, *etc.* to collect various data, and the collected data is delivered to the integrated management server that includes OPC UA client, to be delivered to users.



**Figure 1. Overall Structure of the Testbed to Verify Interoperability**

#### 4.1. Development of OPC UA Server

OPC UA server using OPC UA SDK of Unified Automation is realized in an industrial PC. The industrial PC's OS is Ubuntu, and OPC UA server is composed using C++. OPC UA server is composed to collect various temperature data from FLIR AX8, a thermal camera, through Modbus TCP/IP. The collected data is delivered to OPC UA client via OPC UA server.

OPC UA server using Softing's OPC UA Development Toolkits is realized in an industrial PC. The industrial PC's OS is Ubuntu, and OPC UA server is composed using C++. OPC UA server collects data of temperature, humidity, gas, GPS and vibration sensors connected to the Beaglebone Black, an embedded board, via HTTP. And it delivers the collected data to OPC UA client.

OPC UA server using Kepware's KEPServerEX is realized in an industrial PC. The industrial PC's OS is Windows 7. OPC UA server is composed of 7 alarm lamps and 7 control buttons connected to XGB, PLC of LSIS, through KEPServerEX. The collected data is delivered to OPC UA client via OPC UA server.

OPC UA server using Modigm Server is realized in an industrial PC. The industrial PC's OS is Windows 7. OPC UA server collects data of temperature and humidity sensors connected to XGB, PLC of LSIS, through Modigm Server, and the collected data is delivered to OPC UA client via OPC UA server.

OPC UA server using open62541 is realized in the Humming Board Pro, an embedded board. Humming Board Pro's OS is Debian, and OPC UA server is composed using C as

programming language. OPC UA server is composed to collect various temperature data from FLIR Lepton, a thermal camera. And the collected data is delivered to OPC UA client via OPC UA server.

OPC UA server using node-opcua is realized in Raspberry Pi3, an embedded board. Raspberry Pi3's OS is Raspbian, and OPC UA server is composed using JavaScript and Node.js. OPC UA server using node-opcua is connected to temperature, humidity and gas sensors to collect environmental data, and the collected data is delivered to OPC UA client via OPC UA server.

OPC UA server using FreeOpcUa is realized in Raspberry Pi3, an embedded board. Raspberry Pi3's OS is Raspbian, and OPC UA server is composed using Python.

OPC UA server using FreeOpcUa is connected to temperature, humidity and gas sensors to collect environmental data, and the collected data is delivered to OPC UA client via OPC UA server.

#### **4.2. Development of OPC UA Client**

OPC UA client using Prosys OPC UA<sup>[1]</sup> is installed in an industrial PC. The industrial PC's OS is CentOS, and programming language is Java. And an integrated server that performs functions of OPC UA client is composed to integrate and analyze various data from OPC UA server and deliver such data to users

#### **4.3. Verification of Interoperability**

After operating the testbed, it was confirmed that OPC UA servers composed with different solutions are different in support and realization level of OPC UA's core functions. And the 7 OPC UA servers developed by different solutions performed data transmission between themselves and to 1 OPC UA client without any problems, regardless of development environment. Also, in addition to the developed OPC UA clients, OPC UA client of Unified Automation and Softing performs data transmission between OPC UA server without any problems.

### **5. Conclusion**

This article conducted a comparative analysis on the various current OPC UA solutions, and identified supported core functions and level of their realization. In addition, this article established the testbed and applied OPC UA solutions on various platforms to verify interoperability among heterogeneous devices with OPC UA.

The comparative analysis on solutions to realize OPC UA confirmed that most of core functions of OPC UA are realized, but user must purchase a paid license, and the free trial versions have some limitations. In case of open sources, level of realization of OPC UA is lower than commercial solutions, and there is no open source that realizes all core functions of OPC UA.

The testbed is combined with 4 OPC UA servers and 1 OPC UA client using commercial solutions and 3 OPC UA servers using open sources to integrate and analyze the collected data. Also, in addition with PLC and thermal camera used in industrial sites to collect information, various environment sensors are added in the testbed. After operating the testbed, it was confirmed that OPC UA servers composed with different solutions are different in support and realization level of core functions, but data transmission between them using the established functions is performed without any problems.

This testbed composes the test object based on OPC UA server, but we'll improve it as an interoperability testbed using OPC UA by conducting analysis and realization of OPC UA client through various solutions. Also, based on this testbed, we'll conduct more studies to integrate and link information of industrial device realized with AutomationML and relationship modeling data to OPC UA.

## Acknowledgments

This work was supported by “Development of Open Industry IoT (IIoT) Smart Factory Platform and Factory-Thing Hardware Technology” of Korea Evaluation Institute of Industrial Technology (KEIT) granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (No. 10054486)

This work was supported by the International Collaborative Energy Technology R&D Program of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (No. 20148530050120)

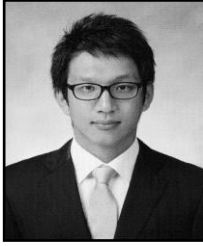
## References

- [1] P. Adolphs, H. Bedenbender, D. Dirzus, M. Ehlich, U. Epple, M. Hankel, R. Heidel, M. Hoffmeister, H. Huhle, B. Karcher, H. Koziolok, R. Pichler, S. Pollmeier, F. Schewe, A. Walter, B. Waser and M. Wollschlaeger, “Reference architecture model industrie 4.0 (rami4.0)”, ZVEI and VDI, Status Report, (2015).
- [2] Industrial Internet Consortium, “Industrial Internet Reference Architecture”, <http://www.iiconsortium.org> [Accessed: August. 07, 2017].
- [3] OPC Foundation, <https://opcfoundation.org> [Accessed: August. 07, 2017].
- [4] D. Van Der Linden, H. Mannaert, W. Kastner, V. Vanderputten, H. Peremans and J. Verelst, “An OPC UA interface for an evolvable ISA88 control module”, In Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference, (2011), pp. 1-9.
- [5] W. Mahnke, S.H. Leitner and M. Damm, “OPC Unified Architecture”, Springer, (2009).
- [6] W. Mahnke, A. Gössling, M. Graube, and L. Urbas, “Information Modeling for Middleware in Automation”, In Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference, (2011).
- [7] PLCopen, <http://www.plcopen.org> [Accessed: August. 07, 2017].
- [8] AIM, <https://www.aim-d.de> [Accessed: August. 07, 2017].
- [9] BACnet, <http://www.bacnet.org> [Accessed: August. 07, 2017].
- [10] FDI, <https://www.fieldcommgroup.org> [Accessed: August. 07, 2017].
- [11] Unified Automation, <https://www.unified-automation.com> [Accessed: August. 07, 2017].
- [12] Softing, <https://company.softing.com> [Accessed: August. 07, 2017].
- [13] Prosys OPC, <https://www.prosysopc.com> [Accessed: August. 07, 2017].
- [14] Kepware, <https://www.kepware.com> [Accessed: August. 07, 2017].
- [15] Modigm, [www.modigm.com](http://www.modigm.com) [Accessed: August. 07, 2017].
- [16] open62541, <https://open62541.org> [Accessed: August. 07, 2017].
- [17] F. Palm, S. Gruner, J. Pfrommer, M. Graube, and L. Urbas, “Open Source as Enabler for OPC UA in Industrial Automation”, In Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference, (2015).
- [18] Github, <https://github.com> [Accessed: August. 07, 2017].
- [19] Mozilla, <https://www.mozilla.org> [Accessed: August. 07, 2017].
- [20] FreeOpcUa, <http://freeopcua.github.io> [Accessed: August. 07, 2017].
- [21] LPGL, <https://www.gnu.org> [Accessed: August. 07, 2017].
- [22] node-opcua, <http://node-opcua.github.io> [Accessed: August. 07, 2017].

## Authors



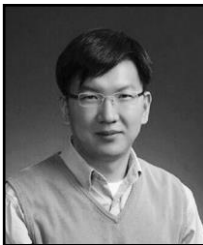
**Byunghun Song**, he is a head of Smart Factory ICT R&D Center at Korea Electronics Technology Institute (KETI), Seongnam-si, South Korea. He received BS, MS and PhD degrees in electrical and electronic engineering from Kwangwoon university, South Korea, in 1998, 2000, and 2004 respectively. He serves as a leader of Infrastructure Team in Korea Smart Factory Foundation (KOSF) and a representative of OPC Foundation Korea. He has over 20 years of experience in the field of wireless sensor network and internet of things. He is now working in the Industrial Internet research center at KETI. His research interests and activities are in areas of Industrial IoT platform and infrastructure architecture for smart factory.



**Hojoong Kim**, he is a Researcher in IoT Platform Research Center at Korea Electronics Technology Institute (KETI). He received BS and MS in electronic engineering from Inha university, South Korea, in 2011 and 2015 respectively. His research interests are in the areas of IoT and Embedded System.



**Wonhee Lee**, he is a Researcher in IoT Platform Research Center at Korea Electronics Technology Institute (KETI). He received BS and MS in electronic engineering from Soongsil university, South Korea, in 2011 and 2014 respectively. His research interests are in the areas of IoT and Embedded System.



**Junho Shin**, he is a Managerial Researcher in IoT Platform Research Center at Korea Electronics Technology Institute (KETI). He received BS and MS in electronic engineering from Ajou university, South Korea, in 1998 and 2000 respectively. His research interests are in the areas of IoT, Sensor Network, and Embedded System.