# A Study on OPC Specifications: Perspective and Challenges

Mai Son
School of Computer Engineering and IT
University of Ulsan
San-29, Moogu-2 Dong, Namgu, Ulsan 680-749, Korea
sonm9285@yahoo.com

Myeong-Jae Yi
School of Computer Engineering and IT
University of Ulsan
San-29, Moogu-2 Dong, Namgu, Ulsan 680-749, Korea
ymj@mail.ulsan.ac.kr

*Abstract*—**OPC (Openness, Productivity, and Collaboration) standards released by the OPC Foundation have provided a solution for system integration in recent years, especially in industrial automation and the enterprise systems that support industry. Because the OPC standards based in turn upon computer industry, they ensure standards technical reliability. The series of specifications contained in OPC standards defined the fundamental mechanisms and functionality of reading and writing process data, monitoring and processing events and alarms, and storing historical data. Many research and development works have proposed to apply them efficiently in the real situation. In this paper, we will not only give readers an overview of the OPC specifications but also try to indicate current situation and problem statement of OPC technology. Afterwards, the new OPC UA standard will be introduced as a good solution to manage and overcome these problems.**

***Keywords: OPC; system integration; OPC Unified Architecture(OPC UA)***

## I. INTRODUCTION

The demand of reusable software components has increased significantly in industrial field, accelerating research and development work. The two main challenges are that how to standardize interfaces between components and how to solve the difficulty of integration among the diverse components and systems [1].

To deal with these problems, OPC technology, the present meaning is Openness, Productivity and Collaboration, has been developed as a data integration middleware for exchanging data between field devices, control systems and other applications. OPC technology based on DCOM not only runs effectively on the Windows but also can be leveraged with XML to operate at non-Windows platforms (OPC XML-DA Specification) [2]. Recently, more than 600 products have been based on the OPC specifications.

Beside its advantages, OPC still had some issues that need to be considered: (i) the non-compatibility with firewall authentication over the internet; (ii) the poor performance of XML Web Services at different platforms. Few approaches have been done on developing design methods or guidelines to overcome the limitation of OPC standard. However, these current approaches have only resulted in small-scale capability or specific applications as studied by Eppler et al. [3], Chilingargyan and Eppler [4], Jia and Li [5], Usami et al. [6], etc.

The OPC Foundation, responsible for development of OPC Specifications, has decided to redesign the key OPC components and technologies with modern, vendor independent solutions. The new specification called OPC Unified Architecture (UA) is being developed and marked an obvious advance to bring Internet-based technology into industrial environment [7]. OPC UA used the approach of binary data encoding to guarantee high performance and XML, web services and Service-Oriented Architecture (SOA) as mechanism for communication [8, 9]. However, one of the biggest barriers to achieve the success of OPC UA is the quite difficult implementation. The OPC Foundation has taken many steps to guarantee that the implementation of the standard would be relatively straightforward and easy process.

This paper makes an effort to provide an overview of the OPC standards before indicating the current problem statement of such standards. The approach based on SOA and new OPC UA specification is proposed as an integrative solution between enterprise systems and the shop floor activities.

This paper is organized as follows: The two next sections provide an overview of the OPC technology with its specifications. In Section 4, the current situation and problem statements of the OPC standards are presented. Thereafter a discussion will be proposed in Section 5. Finally, the conclusion of the work is written in Section 6.

## II. OPC OVERVIEW

OPC has emerged as the worldwide industrial standard based on Microsoft's Distributed Component Object Model (DCOM) in recent years. The standard is not only capable of connectivity among automation components with control hardware and field devices but also provides the interoperability of Office products and information system on the company level such as Enterprise Resource Planning (ERP) and Manufacturing Execution System (MES). An example of the difference between conventional communication architecture and communication architecture based on the OPC standard is illustrated in Fig.1 and Fig.2.


978-1-4244-9037-0/10/$26.00 

With the former, each application has to install a driver for each device in order to be able to communicate with each other. The number of different devices increases, leading to the multitude of attached drivers. To exceed this limitation, OPC technology supported a standard mechanism for the communications among various data sources, devices on the plant floor, or a database in a control room.

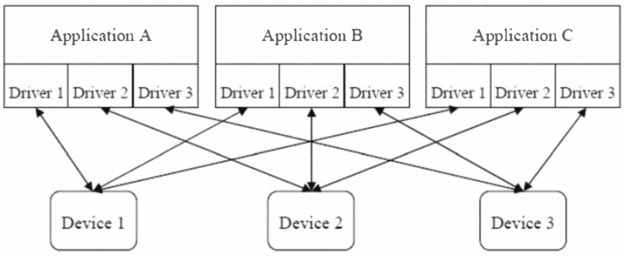


Figure 1. Conventional communication architecture

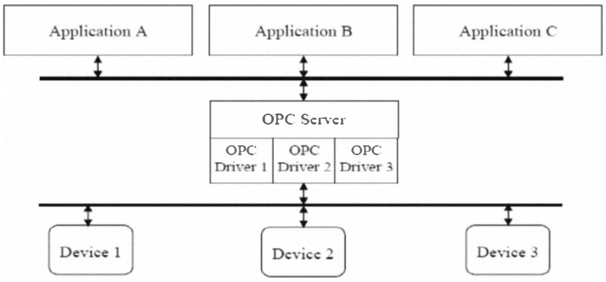


Figure 2. Communication architecture based on the OPC standard

Based on the client-server architecture, OPC server provides a standard interface to the OPC COM objects, which let OPC client applications exchange data as well as control commands in a generic way. OPC client applications can connect with one or more OPC servers from different suppliers.

## III. OPC SPECIFICATIONS

OPC is a series of specifications, created through the collaboration of a number of leading worldwide automation hardware and software suppliers, defined as sets of standard objects, methods, and according to different requirements within industrial applications. The initial OPC specification (version 1.0) was released in 1996.

These interfaces allow a highly efficient data exchange between software components of different manufacturers. This section will briefly describe an overview of four major specifications:

- OPC Data Access (DA) Specification
- OPC XML-DA Specification
- OPC Historical Data Access (HDA) Specification
- OPC Alarm and Event (A&E) and Batch.

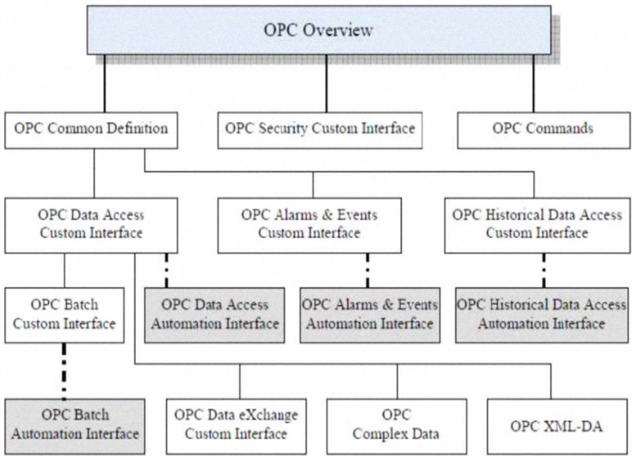


Figure 3. Overview of OPC Standards

### A. *OPC Data Access (DA) Specification*

This OPC standard defines a set of standard COM objects, methods, and properties between client and server for reading, writing and monitoring the variables containing current process data. It intends to standardize the mechanism for communicating a numerous of data sources, whether they are hardware I/O devices on the plant floor or databases in control rooms.

All the different data sources, e.g., temperature sensors, and data sinks, e.g., controllers are organized and made available by a server's namespace structure [1]. Additionally, DA Server offers methods to navigate through the OPC object hierarchy.

In the first step, an OPC client need to establish a connection to a server by creating *OPCServer* object which is the top level object of the hierarchy. The OPC clients can access to the data at one or more OPC servers from different suppliers, whether data are coming from an OPC connected to a PLC (Programmable Logic Control system); industrial networks such as FOUNDATION Fieldbus, PROFIBUS, or DeviceNet; SCADA (Supervisory, Control, and Data Acquisition system), and so forth.

The next step, the client groups the OPC items with identical settings such as update time in an *OPCGroup* object. Finally, in the real application, several values are read and written at the same time, so it is more efficient to manage several *OPCItem* objects with a single call via methods provided by *OPCGroup* object.

The communication mechanisms are used for exchanging data between an OPC client and server such as *synchronous calls, asynchronous calls, refresh,* and *subscription.* With synchronous reading, for example, the client has to wait for the return value after calling the method whilst the client with the asynchronous calling will immediately get feedback. The refresh and subscription are *callback mechanism* used to access predefined sets of data points on the plant floor.

978-1-4244-9036-3/10/$26.00  

OPC Data Access, the crucial specification, is implemented in 99% of the products using OPC technology today [8]

*B. OPC XML-DA Specification*

OPC XML-DA was defined as an effort to solve the main problems of OPC COM DA:

- OPC COM DA, based on COM/DCOM, was only implemented successfully in Windows platform but in a heterogeneous environment, computers cannot all be expected to implement the corresponding object models.
- OPC COM DA defined callback mechanism. This link between clients and servers did not exist because HTTP is a stateless protocol

OPC XML-DA defined as independent specification in order to guarantee better interoperability with non-windows platforms and more flexible internet access. It substituted HTTP/SOAP and Web Service technologies for classic COM/DCOM. The set of methods using for exchanging data was reduced minimum with only eight services and made suitable for stateless environment. However, The XML transmission between clients and servers is less efficient than the transmission of binary data with DCOM.

*C. OPC Historical Data Access (HDA) Specification*

Many plant decisions are based on historical data: failure prediction, drift prediction, root cause analysis, and wear-tear, etc. With respect to historical data, OPC HDA Servers enable clients to access two basic types of data: (i) raw data, which means the historical data stored and (ii) aggregated data, which means data extracted from the raw data [1]. Depending on different implementation, OPC HDA Servers are divided into two kinds.

- Simple trend data servers which only implement some optional interfaces and access to raw data
- Complex data compression and analysis servers that allow for data processing such as analysis of data, e.g., average value, minimum value and maximum value, regeneration of data, addition of annotations, and reading histories of data changes

OPC HDA defined namespace contains all the historical data, to which access is provided by the server and object hierarchy as well. Unlike the DA Server, there are no objects comparable to the *OPCGroup* and *OPCItem* objects. The client directly addresses data items by means of the handle without creating objects for this purpose in the server.

*D. OPC Alarms & Events (A&E) Specification*

This standard provides flexible interface for alarm and event notifications on demand (in contrast to the continuous data flow of OPC Data Access). These notifications include process alarms, operator actions, informational message, and tracking/auditing messages [11]. Before transmitting process alarms and events, OPC A&E client and server need to establish a connection by creating an *OPCEventServer* object. An *OPCEventSupcription* will be generated to receive the event message in the next step. The quantity of the return events can be reduced by setting certain filter criteria, for example, filter by event types, by priority, or by event source.

The OPC AE server can be implemented as an individual component that obtains information directly from the physical devices and applications or from a component together with OPC DA Server

*E. OPC Batch Specification*

Unlike the specification Data Access, Alarms and Events, and Historical Data Access, the OPC Batch Specification does not define any completely new programming interface between a client and a server. It defines supplement to the Data Access Specification for the case of batch processing [1]. When executing a bath process, recipe data are sent and report data are received, respectively. The solutions are concerned for visualization, report generation, process control, and devices executing control sequences and generating report data.

In principle, OPC Batch servers enable clients to access batch information. The IEC standard distinguishes four types of batch information that enable clients to access such as equipment capabilities, current operating conditions, historical contents, and recipe contents.

## IV. The Problem Statement of OPC Technology

*A. COM/DCOM*

Almost OPC specifications are based on COM/DCOM technology made available by Microsoft and defined transparent interaction mechanism between distributed components including data objects and application objects. The advantage of this approach was the reduction of the specification work, the strategic vision leading to the OPC success. Unfortunately, the classical OPC technology inherited some limitations from such a Microsoft's technology which is difficult, if not possible, to resolve: (i) COM is only supported in Windows platforms, it is not easy to find a reliable COM implementation on non-Windows platforms; (ii) DCOM can be used to the applications over the Internet, but firewall authentication problems are not easy to set up; (iii) data exchange between devices on the plant floor and the enterprise application such as MES and ERP is an issue that needs to be tackled.

*B. XML Web Services*

XML Web services provide a viable solution for enabling data and system interoperability. XML Web services use XML-based messaging as a fundamental means of data communication to create the bridge between different systems, and programming languages. OPC XML-DA, developed based on XML Web Services, is a best way to exchange data cross-platform. But due to its high resource consumption and the limited performance of the large size of XML messages, it was


978-1-4244-9037-0/10/$26.00

not as successful as expected for this type of application. However, binary data encoding approach can be used for XML web services to improve the performance [12, 13].

### *C. OPC Unified Architecture (UA) Standard*

OPC-UA (OPC Unified Architecture) extends the highly successful OPC communication protocol, used in horizontal integration where information is transferred between automation system and in vertical integration between different layers of factory automation [7].

#### *1) Information Modeling*

The fundamental components of OPC UA are transport mechanisms optimized for different use cases and data modeling enhanced from Classic OPC. Modeling information is an important issue in order to design and develop information system applications. OPC UA provides more powerful possibilities exposing the semantic of the provided data. It covers all successful features known from Classic OPC and exposing much more semantics which allow clients to process highly sophisticated tasks.

Classical OPC has fairly a simple data model; each specification handles a different aspect of data. With OPC UA, all data are handled in standard way. This approach reduces the time and cost for application development. A unified object model for three types of data can be described as follows: (i) current and historical data can be stored in variables; (ii) the commands to control devices at the process plants can be considered as method services for execution; (iii) the occurrence of an alarm or event from hardware devices can be considered as an event service

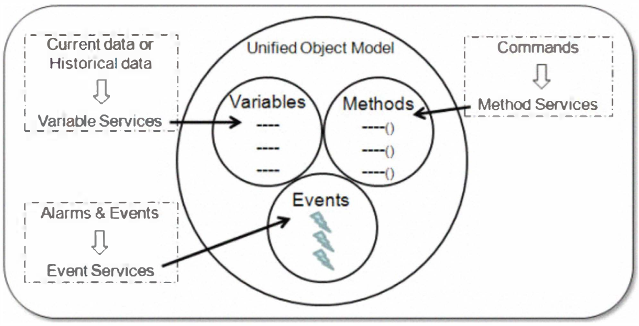


Figure 4. Unified object model for current data, historical data, alarms and events, and commands

#### *2) Multiple platforms*

Classical OPC technology was developed based on COM/DCOM technology running efficiently on Windows but not other operating systems. To resolve this issue, OPC UA support multiple platform ability, leading to the fact that applications can developed on non-windows platform such as Linux and embedded devices [14].

#### *3) Internet access*

OPC UA uses Web Services instead of DCOM for data transportation. Consequently, OPC UA inherited many advantages from Web Services, especially firewall-friendly communication between networks and independence from specific operating systems

#### *4) Complex data structures*

The existing specifications only offer a simple hierarchical organization of time, whereas OPC UA offers Meta information models that can be easily extended. A vendor, for instance, can use base models and extend them with vendor-specific information about its devices or integrate third-party devices easily and seamlessly

#### *5) Security mechanism*

Unlike classical OPC depended on security of COM/DOM, OPC UA provides a flexible security model that allows OPC UA applications to be run at different levels in the automation pyramid and at same time meeting the security requirement for each environment [8].

#### *6) High performance*

To ensure high performance, the UA Binary web services are being planned to attach the UA messages encoded in UA Binary to a SOAP message that greatly increases speed and efficiency compared to the XML/text encoding.

## V. Discussion

#### *1) OPC-UA New Integration Technology*

The limitation of Classical OPC specifications leaded to the new potential standard for data communication in process automation, OPC Unified Architecture (OPC UA). The OPC UA standard intends to enable enterprise interoperability and expect to solve the complex exchange problems between platforms. In addition, OPC UA Software Development Kit (SDK) was introduced as an effort to reduce the development time and facilitate faster interoperability for an OPC UA application but it is still incomplete version. In addition, the .NET, ANSI C, and Java implementation are being developed and maintained by the OPC Foundation

#### *2) Migration Path*

To ensure an easy way to migrate from an OPC/DCOM based application infrastructure into a new OPC UA/SOA based one, the OPC UA wrapper and proxy components are developed.

OPC UA Wrappers are used to allow OPC UA clients to access Classical OPC servers by using an OPC COM client interface. It maps OPC UA request to corresponding COM calls and COM return values to corresponding OPC UA responses. In the contrary, OPC UA proxies use COM server interface to map COM calls to corresponding OPC UA requests and OPC UA responses to COM return values.

Despite the benefits, many innovations and advantages of the OPC UA technology are lost such as the uniform access to process data, historical data and alarms in one server address space, programs, type information, and structured data types

### 3) *Device Profile for Web Service*

The OPC UA standard addresses communication between devices and higher-level systems rather than peer-to-peer communication between devices. Therefore, the Devices Profile for Web Services (DPWS) was defined as the solution assisting SOA paradigm in interconnecting devices and making them available to IT systems [10]. The integration of field devices into the business-IT systems through the SOA approach and the OPC UA is a promising approach in the future

The DPWS includes WS-Discovery for plug-and-play device discovery and WS-Eventing for publish-subscribe asynchronous event notification.

### 4) *New challenges*

OPC UA is making new challenges to researchers such as (i) new device integration concept, (ii) new scenarios like Asset Management and Manufacturing Execution System (MES), and (iii) the design and implementation of this new standard for automation system applications [15].

## VI. CONCLUSION

This paper gave you a brief overview of the OPC technology with some crucial specifications. Although OPC technology with plug-and-play capabilities gained some advantages including reduced training costs, custom development cost, and long-term maintenance cost, the binding to COM/DCOM brought about several drawbacks such as:

- The dependence on the Windows operating system;
- The frequent problems in the configuration of DCOM;
- The performance of these approaches limited by using pure XML textual data representation.

Therefore, OPC Unified architecture was released as a new standard overcoming the challenges of traditional OPC technology.

Since industrial systems need to be independent of any operating systems and platform, OPC UA with the combination of SOA paradigm and web technologies will step by step replace the OPC application based on DCOM technology and become a good choice for the development of web-enabled industrial automation and manufacturing software system as well as for the enterprise integration.

## ACKNOWLEDGEMENTS

This work was supported in part by the Korean Ministry of Knowledge Economy and Ulsan Metropolitan City through the Network-based Automation Research Center (NARC) at the University of Ulsan and by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MEST) (NRF-2009-0076248).

## REFERENCES

[1] F. Iwanitz, J. Lange, "OPC Fundamentals, Impementation, and Application", 3rd ed., Huthig Verlag Heidelberg, 2006

[2] Q. Jiaxin, "Application Research on Enterprise Integrated Automation Based on OPC & OPC-XML", Proceedings of the IEEE International Conference on Automation and Logistiecs, August 2009, pp. 747-751

[3] J. Boyer, M. Hughes, J. Reagle, "XML Signature XPath Filter", Version 2.0, 2002. http://www.w3.org/TR/2002/REC-xmldsig-filter2-20021108/

[4] S. Bandini, A. Mosca, M. Palmonari, F. Sartori, "A Conceptual Framework for Monitoring and Control System Development", UMICS 2004.LNCS, vol. 3272, pp. 111-124, Springer, Heidelberg (2004)

[5] S. Chilingargyan, W. Eppler, "High Speed Data Exchange Protocol for Modern Distributed Data Acquisition Systems based on OPC XML-DA", Proceedings of the 14$^{th}$ IEEE-NPSS Real-time Conference, pp. 352-256

[6] M.E. Frayad, D.C. Schmidt, R.E. Johnson, "Building Application Frameworks: Object-Oriented Foundation of Framework Design", Wiley & Sons, Chichester (1999)

[7] L.M.S. de Sourza, P. Spiess, D. Guinard, M. Kaohler, S. Karnouskos, D. Savio, "A Web Service Based Shop Floor Integration Infrastructure", IOT 2008.LNCS, col. 4952, pp. 50-67, Springer, Heidelberg (2008)

[8] W. Mahnke, S-H. Leitner, M. Damm, "OPC Unified Architecture", Springer, 2009

[9] The OPC Foundation, The OPC Unified Architecture Specifications: Parts 1-11, Version 1.xx (2008)

[10] E. Zeeb, A. Bobek, H. Bohn, F. Golatowski, "Service-Oriented Architectures for Embedded Systems Using Devices Profile for Web Services", Proceeding of the 2007 21st International Conference on Advanced Information Networking and Application Workshops, pp. 956-963

[11] P. Bhatt, R.P Baldevia, "Integrate IEDS with OPC Technology", Proceedings of 2006.PS '06 on Power Systems Conference: Advanced Metering, Protection, Control, Communication, and Distributed Resources, pp. 391-407

[12] W. Eppler, A. Beglarian, S. Chiligarian, S. Kelly, V. Hartmann, H. Gemmeke, "New Control System Aspects for Physical Experiments", IEEE Transactions on Nuclear Science 51(3), 2004, pp. 482-488

[13] S. Chilingargyan, W. Eppler, "High Speed Data Exchange Protocol for Modern Distributed Data Acquisition Systems Based on OPC XML-DA", Proceedings of the 14th IEEE-NPSS Real-time Conference, 2005, pp. 352-356

[14] H. Lu, Z. Yan, "Research on key technology of the address space for OPC UA Server", Proceedings of 2010 2nd International Conference on Advanced Comuter Control (ICACC), March 2010, pp. 278-281

[15] V.V. Tan, D.-S. Yoo, M.-J. Yi, "Device Integration Approach to OPC UA-Based Process Automation Systems with FDT/DTM and EDDL", ICIC 2009, LNAI 5755, pp. 1001-1012

 978-1-4244-9037-0/10/$26.00 