2013 XXIV International Conference on
Information, Communication and Automation Technologies (ICAT)
October 30 – November 01, 2013, Sarajevo, Bosnia and Herzegovina

A Survey on OPC and OPC-UA

About the standard, developments and investigations

Schwarz M.H.
Safety Computer Technology
University of Kassel
Kassel, Germany
m.schwarz@uni-kassel.de

Börcsök J.
Computer Architecture und System Programming
University of Kassel
Kassel, Germany
j.boercsoek@uni-kassel.de

***Abstract*— In 1996, a new standard was announced that should serve as a software interface to exchange process data and to solve the problem to exchange process data using different industrial protocols and communication systems. A successful story started since then with few additional standards like the Alarm and Event standard using the OPC approach and some revisions and new editions. Ten years later a new approach was created that unified all existing standards and was also concerned with e.g. interoperability, security and web-based systems. This paper details the different OPC standards, tries to answer the question why this standard is important for industries and academia and where current research and development utilising those standards.**

Keywords – OLE for Process Control, OPC, Unified Architecture, OPC-UA, Openness Productivity Collaboration

I. INTRODUCTION

In the mid 1980's industries and academia started to investigate, develop and establish networks, protocols and bus systems [8],[9],[30]. Within few years over 50 different bus systems exist [30], few became universal communication systems in industries, some are used in some specific areas and some vanished. For example, *Profibus* and its derivates are often used in process industries, *CAN*-bus is mostly used in cars and office networks are often based on Ethernet. In a system different devices have to exchange information like a *Human Machine Interface* (HMI) and *Supervisory Control and Data Acquisition* (SCADA) system; those have to collect, analyse and display data from various devices using different protocols and networks [14]. This task becomes non-trivial and fault-prone. For each device a software driver has to be provided by the device-company, a schematic is shown in Fig 1. Changes in the protocol specification resulted in malfunctioning communications and adjustments had to be made, which was time consuming and expensive [30]. On the one hand this kept competitors away on the other hand it was difficult to get into new areas.

In 1995, companies like Fisher-Rosemount, Intellution, Intuitive Technology, Opto22, Rockwell, and others formed a task force to find a solution to this problem [8],[9],[11],[14]. Within a few months a new standard was established, which describes server-client software architecture to collect real process data from devices to pass them e.g. to SCADA systems. The first specification was launched in August 1996 as the *OPC-DA* 1.0 (Data Access) and the OPC foundation was founded to continue, maintain and market the standards.

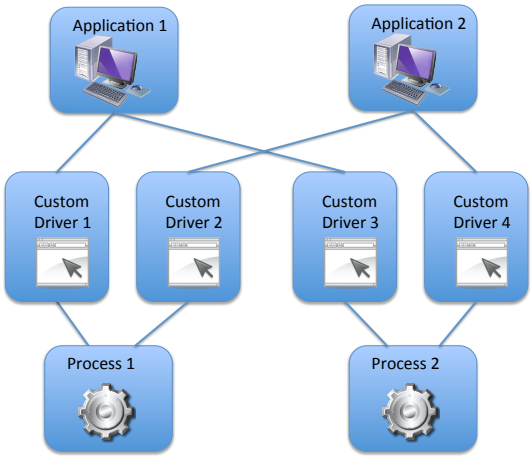

Fig. 1: Communication without OPC

The original name of OPC stood for ***O****LE for* ***P****rocess* ***C****ontrol*, as the client-server architecture was based on Microsoft's OLE (Object Linking and Embedding), a technology to develop object orientated structures, which was later combined in COM (Component Object Model) and DCOM (Distributed component Object Model) [3],[8],[9],[11]. To base the client-server architecture on Microsoft's components was probably one of the reasons for the giant success of OPC [3],[11]. Several different standards followed and will be detailed in the next section.

Although OPC is accepted in industries and has been widely used in many applications, some negative aspects arose [3],[8],[9],[14] during the years. Operating OPC on other operating systems such as Linux had the problem that COM/DCOM functionality had to be emulated to be used which has caused many problems [14]. In the mid 1990s, security aspects were not being seen as a big issue and should be handled by the operating system [2],[3]. This has been indicated as a wrong assumption, a separate specification on

978-1-4799-0431-0/13/$31.00 ©2013 IEEE

security was launched in 2000, but had virtual no effect on the market [3].

Data belonging to each other but were provided by different OPC server architectures could not be grouped together. For example, the process generated an alarm, because the process value was larger as allowed, then the alarm was provided by the *Alarm and Event* (A&E) server, the actual process value by the *Data Access* (DA) server and that information could not be assembled together [3],[8],[9],[11], [14].

Although, the different OPC client-server architectures especially *DA* and *AE* were and still are highly accepted in process and automation industries, in 2006 it was time for a new step to create a new concept that combined the different separate standards into one, that uses one address space for all different OPC client-server architectures instead for each architecture a separate address space. The new standard also integrates security aspects and reliable characteristics like fault tolerance, redundancy, interoperability and it uses web-based technologies [8],[9],[11],[14].

With the new standard name *OPC-Unified Architecture* (OPC-UA) announced in 2006 and a revision with additional parts of the standard in 2009, the previous OPC versions were renamed in *Classic OPC*, as many companies and manufacturers still provide and develop many OPC architectures and systems [11],[14].

The OPC Foundation also went a different root with the new OPC-UA standard [11] compared to the *Classic OPC*. The *Classic OPC* is a de-facto standard accepted in process and automation industry but not approved as a standard like the IEC 61508 [7] or IEC 61131 [6]. Therefore, the OPC Foundation got involved the IEC organisation at an early stage and in February 2010 the first two parts of the OPC-UA specification became also an IEC standard, other parts followed and will follow. The OPC-UA standard is since then also available as IEC 62541 [11],[3],[14]. Furthermore, the acronym OPC (OLE for Process Control) started to change into *Openness, Productivity Collaboration.*

Additionally, the OPC foundation started to cooperate with others, one example is the PLCopen [29] to develop a common model to exchange process variables.

After the history of the de-facto standard OPC (Classic OPC) and the standard OPC-UA (IEC62541) has been roughly detailed, the outline of the remaining paper is as follows: Section 2 will describe the different specifications of the *Classic OPC*. Section 3 will detail the main innovation of OPC-UA. Section 4 will present a survey on on-going research and investigations using or enhancing OPC and OPC-UA. Section 5 will draw some conclusions.

II. CLASSIC OPC

This section describes the different OPC specifications that are summarised under the umbrella of *Classic OPC*. All have in common that they are based on client-server architecture. They are definitions of common interfaces that permit applications using OPC client and OPC server to exchange data, events and information with devices as shown in the figure below [33].

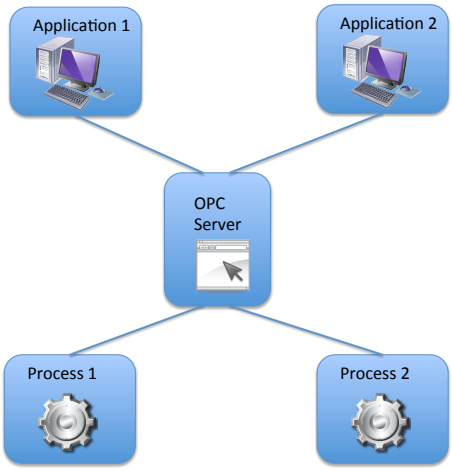

Fig. 2: Communication using OPC

A. Data Access (DA) Specification

The specification of *OPC Data Access* was the first one published in 1996 and was the starting point of the OPC foundation [8],[9],[11],[14]. The current version is 3.0 published in 2003 [22]. The de-facto standard proposes a client server architecture to exchange real-time process data in a specified format, as shown in Table 1.

TABLE 1: OPC Items and Attributes

Variable	**Attribute**
Process Value	*Value*
Time stamp	*Time/Date*
Reliability	*Good/unknown/bad*

The *DA*-server can normally deal with several clients and they can book required items consisting of the three attributes presented above. When the device itself is not able to provide any time stamp then the server has to provide it. If the link to the device is disconnected then the server can indicate this with a reliability flag and the client can react on this fact [22].

B. Alarms and Events (A&E) Specification

The *A&E* specification [19] defines an interface for server and clients to exchange information on alarms, events and their acknowledgements. The server receives process information from devices and analyses them and provides the resulting events or alarms with the necessary information. It is important that the *A&E* server evaluates basically the same process values as the *DA* server but to get the information of the e.g. alarm and the actual process value two servers are necessary, the *DA* and the *A&E* server and the client has to be connected to both.

C. Historical Data Access (HDA) Specification

The *Historical Data Access* [24] specification can be seen as an extension of the *Data Access* specification. It deals with process data, but not with real-time data and the purpose of this specification differs. The process data are stored and can be accessed as either raw data or aggregated data. From the stored, past (historical) data, trends, characteristics, mean values, minimum, maximum etc. within a specified time span can be calculated and sent to the client. The purpose of this client server architecture is more to use the past data for e.g. optimising a process or evaluating the quality of products [8],[9],[11],[24],[33].

D. Commands Specification

Sometimes, it is not enough to read and write data values or to get informed about an event which occurred, but initiates commands to be executed to control or configure a device or system or to start a programme reload. Therefore, the *Commands* specification was created to provide an interface for executing defined commands [3], [21].

E. XML Data Access Specification

The *XML Data Access* specification is based on the OPC *Data Access* specification but uses XML and Web-services without any COM/DCOM communication, so that platform independence can be achieved [8],[9],[11],[14],[26]. However, the standard can be seen as the predecessor of OPC-UA, but were released too late. Customer, manufacturers and developer were waiting for OPC-UA and not keen in developing a client-server architecture that would be soon overtaken by the new OPC-UA specification [11],[14].

F. Data Exchange (DX) Specification

The purpose of this specification is to use again the DA interface for exchanging data but between servers. It is a horizontal communication for e.g. to provide a back up or redundant server strategy. Another possibility would be to incorporate a client in the server that communicates with the second server and vice versa, which is shown in the figure below [11],[14],[23].

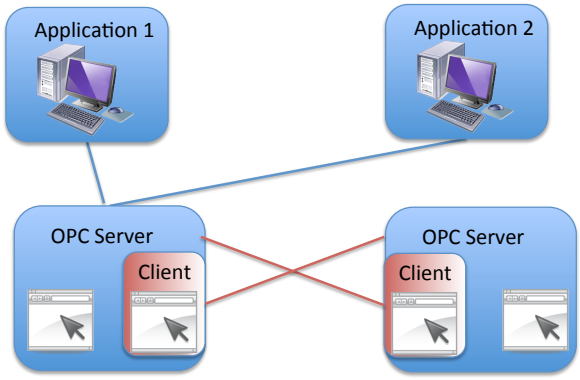

Fig. 3: OPC server communication using an inherent client

G. Batch Specification

The OPC *Batch* specification [8],[9],[11],[20],[33] is not an entirely new interface, rather an extension to the *Data Access* specification for the special case of batch processes. A batch process consists of different formulas and recipes to fabricate or produce products. Within the execution of the batches, devices have to communicate and exchange information. Subscription data is sent and report information is received. Products for batch processing have to be manufactured according to the IEC 61512-1 [8],[9],[11],[33]. This includes the visualisation, report generation, sequence control systems and equipment. Between these components and products, information about the properties of the equipment, current working conditions, historic data and substances, volumes and capacity of the batch have to be exchanged. The OPC specification supplies interoperability between different components, equipment and system of the batch processing industries. Therefore, this specification does not describe a solution for batch regulation problems, but solutions of different manufacturers in a heterogeneous environment [8],[9],[11],[25],[33].

H. Summary

So far, the different specifications of the de-facto standard have been detailed and can be graphically presented as follows:

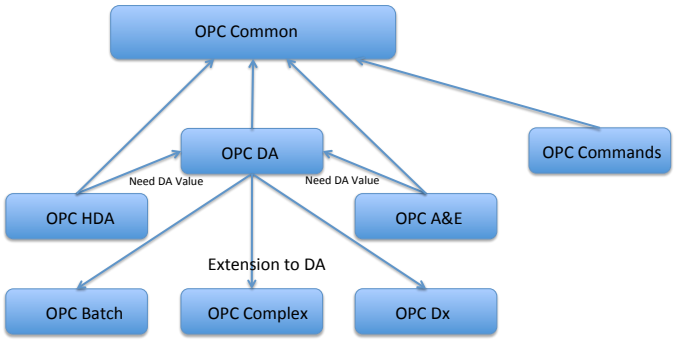

Fig. 4: Relation of the different OPC specifications

Not all specifications are shown in the figure above, but it demonstrates how important the OPC *DA* specification is, which is also an indication why several revisions exists of this specification and why the others have not been altered that much.

The next section deals with the new OPC specification, which is also an IEC standard, and not a de-facto standard as the classic OPC.

III. OPC Unified Approach

Classic OPC (which was state of the art at its appearance) has been widely accepted in automation and process industries and is still a de-facto standard. The different specifications detailed in the previous section were necessary and resulted from demands from industries. After roughly 10 years after the first specification was released a new, combined standard arose. This was necessary as the different individual standards defined different address spaces and those could not be merged together even if the same variable with its aggregated values was used, the scenario is shown in the figure below [3],[11]. Security and interoperability were not issues at the time when the first de-facto standard was released; it was

assumed that this should be handled by the operating system, which was a mistake [2]. Web-based systems and XML were in the mid 1990's no issues.

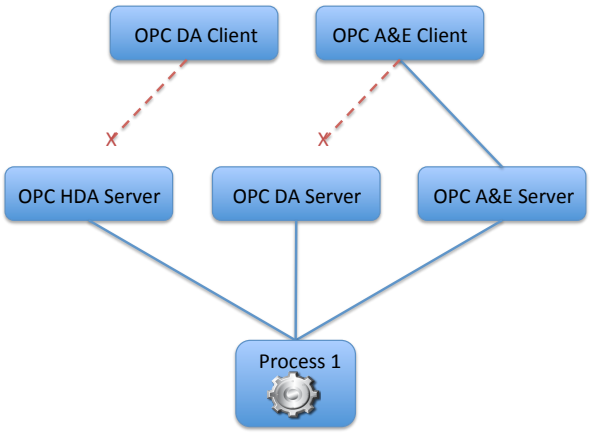

Fig.5: OPC communication only within the same specification possible

One major step was to unify all different address spaces to one that the OPC-UA server provides one service for the object to be accessed by the client. The standard characterises the object as variables, events and methods as shown in the figure below. *Variables* are related to *OPC-DA* and *OPC-HDA*, *Events* to *OPC-A&E* and *Methods* to *OPC-Commands*, when it would be compared to the *Classic OPC* [3],[11].

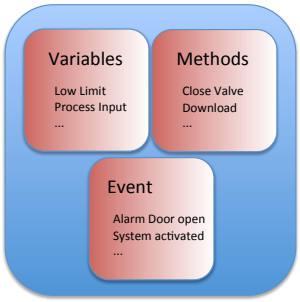

Fig. 6: OPC –UA Object model

A. *Security*

As mentioned earlier, security is an objective that has been underestimated in the past [1] and when it was recognised as a severe problem then the security specification was announced too late to be accepted and adopted in industries. One workaround is to tunnel the information from the client to server [10],[16],[18],[33].

OPC-UA Security is considered at two different stages of the communication, one is at the application layer the other one is at the communication layer. The application layer has to deal with the user authentication and user authorisation. A secure channel is provided at the communication layer, which is used by the application layer to pass the date from client to the server. The communication layer includes confidentiality, integrity and application authentication [3],[11],[27].

User authentication is to identify a e.g. client, by providing a password, an X.509V3 certificate or a WS-Security token [3],[11],[27].

User authorisation [3],[11],[27] defines individual rights to access certain services or denial certain services. The specification itself does not state how the user should legitimate its rights, but provides the possibility that this can be implemented and applied.

Confidentiality is to secure the transmission of the data from the server to the client, therefore, the request and responses have to be encrypted [3],[11],[27] so that other cannot read easily the information that was sent.

Integrity is to ensure that the message is not altered during transmission. Signatures can be used to ensure that the client received the identical information the server had sent [3],[11],[27].

Also at the top level, the application level, the client and server application should identify them, using certificates, which should be done during the time when the secure channel is established. Therefore, the application can decide either to accept or reject the request and responses [3],[11],[27].

Figure 7 shows a schematic of the OPC security, where the application authentication is done via the secure channel, when it is established.

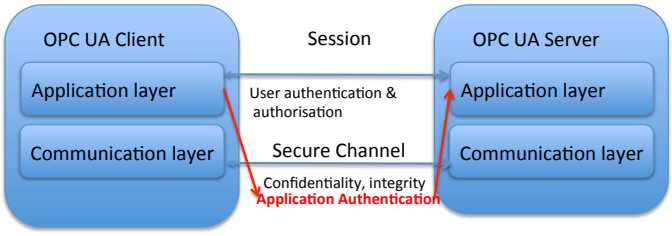

Fig. 7: Security model for OPC UA

B. *Communication Stack*

The transmission of data can be done in three different ways; those are: Native Binary, XML Web-services and SOAP/HTTP with Binary. The figure below shows the three different ways.

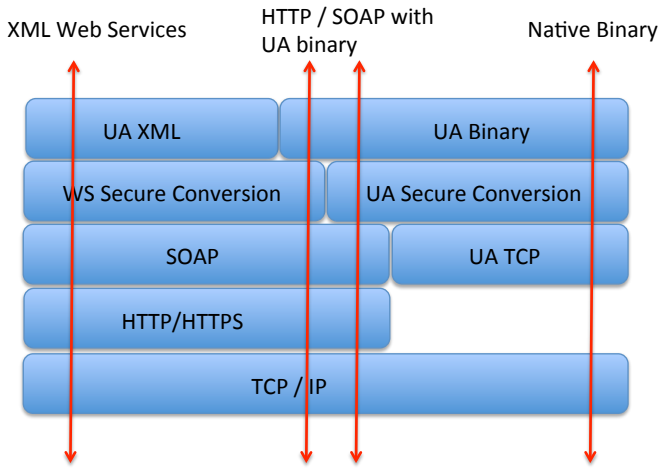

Fig. 8: Communication stack

With this communication stack the OPC-UA approach provides an interoperability layer that rely on no specific operating system or protocol [3],[11],[28].

Both, UA-XML and UA-Binary are securing and transmitting the requests and responses. While the *UA-XML* relies on encoding schemes for Web Services, the UA-Binary on binary formats as specified in OPC-UA part 6 [28] for high speed communication or embedded systems, where communication using XML would require a high amount of resources like memory and CPU power, which is often too much for embedded devices. The security conversion uses either standard protocols used by Web Services Security or the binary version defined in the specification part 6 for a high performance communication in embedded systems, the same two different ways are provided for the transport mechanisms. Also a mixture of the two XML and Binary is possible [3],[11],[28].

IV. Survey and Investigations

This section describes few interesting and relevant on-going research areas about OPC and OPC-UA. The first part deals with the classic OPC, the second part concentrates on OPC-UA.

A. Classic OPC

Few interesting introductory papers exist, such as the paper by Mai Son & Myeong-Jae Yi [34] or by Liu et al. [13], which describes the viewpoints and tasks of the classical OPC and the unified architecture. The first provides an interesting summary of the different specifications and problems to be solved the second describes the development of an OPC system to be used in Distributed Control System (DCS).

Classical OPC did not get much interest in research communities; it seemed that this has changed, when OPC became interesting to be connected to the Internet. A development and investigation is using a distributed OPC architecture for remote control and monitoring [32]. The communications of the different local processes are using PLCs and OPC servers to monitor and control those processes on a local PC. Additionally, a web-based communication was added in such a way that different control stations via the Internet could remotely control those processes. The authors neglected the DCOM interface and developed an Internet interface, which is a database at a web-server to exchange the current (and past process data), which can be seen as an adoption of OPC-UA without utilising this standard [32].

Several papers exist, that using OPC-XML for their investigations. The first investigation [36] uses an OPC-XML-DA server to transmit process data via the Internet to be used for example for production management and enterprise applications such as MES (Manufacturing Execution Systems) ERP (Enterprise Resource Planning) CMMS (Computerised Maintenance Management Systems) and others. The OPC-XML server inherits an OPC-Complex-Data architecture to read, send and write complex and various data types such as abstract elements, integer or strings. Additionally, the authors propose a method using XML for security purposes, as this is an important issue, when communication is done via the Internet [36].

In another research [38], a procedure is described that uses an OPC-XML-DA server and translate the data into a SOAP compatible message that can be transmitted via the Internet. This method can be interpreted as an OPC-UA method mapped onto Classic OPC.

B. OPC Unified Architecture

OPC is a new and combining technology, built on a proven client and server architecture [11],[14],[35]. Changing to a new technology is not a trivial task especially in industries. Customer who uses classical OPC might be not keen in changing the technology, at least not completely at once. Existing systems have to be included without a complete reinstallation. The different ways to wrap the older OPC clients and servers into a new OPC-UA compatible framework has been investigated and described [3],[4],[11]. In such a way those can be used and accessed by OPC-UA systems.

OPC-UA is not only developed for the communication with production management and enterprise applications but also for PLCs and process devices. In one application the authors demonstrate that OPC-UA ANSI C stack can be ported onto an embedded system with non-windows based operating system [5].

A successful application utilising OPC-UA with Modbus is presented by the authors of [37], where a strategy is described to control and monitor a large air conditioning system. This paper demonstrates how multilateral UPC-UA can be used, with different communication protocols.

OPC-UA is not only a client-server architecture for automation and process industries. The investigation by Maka et al. [15] demonstrates the multilateral usage. The object orientated structure is developed for an information system to give customers using public transportation real time information and to provide for the traveller the optimal route and to identify problems for example that a bus is delayed or to determine the current usage rate [15].

Security and safety issues are important, as automation and process facilities should be operated with standard PCs and standard software. Therefore, issues that are discussed in the area information technologies (IT) are becoming issues in process and automation industries as well. Several investigations exist on this topic.

OPC-UA has a security model included as described in the previous section; however, as stated this might be not enough to defend all threats [31], which are known for attacks in the Internet and for web-services. In this investigation [31] a security strategy is presented and built-in OPC-UA security strategy. A module has been proposed that can be configured to the different security requirement levels.

When using OPC-UA XML Web-services for transferring process data, then web browser would be an interesting possibility to monitor and display the data [1]. However, the web browser, which accept and uses XML functionality has also to operate security features in order to be accepted in industries. The cipher and decipher algorithms have to execute

complex mathematical functions, which was not the original intention of XML. The authors present methods in JavaScript to improve and overcome this situation [1].

The remaining two papers show that OPC-UA also gets attention of other organisations. As mentioned in the previous section the OPC foundation cooperates with PLCOpen. In the first paper [17] a method is described to translate IEC 61131-3 software models into the information model of OPC-UA. Again, XML is the key for this conversion. This paper also demonstrates that OPC-UA has become much more than a simple client-server specification.

The last paper shows that the OPC-UA model can be used as an alternative model for the IEC 61850 for *Smart Grid Automation*. The results proposed by the authors [12] provide several advantages, also in security and communications and shows the usability of the OPC-UA specification.

V. CONCLUSIONS

This paper provides an introduction to *OPC* (OLE for Process control) and *OPC Unified Architecture*. The meaning of the acronym OPC has changed within the last years to *Openness, Productivity, Collaboration*. Interesting and important investigations and research using and enhancing OPC were presented. The new OPC-UA specification has also become an international standard – IEC 62541 and attracts research and different organisations as the PLCOpen. The collected and briefly described investigations show that OPC-UA is not only an area for industries but leave a lot of space for academia as well, such as security, embedded systems or interesting applications.

REFERENCES

[1] Braune A., Henning S., Hegler S., Evaluation of OPC UA Secure Communication in Web Browser Applications. (2008). The IEEE International Conference on Industrial Informatics (INDIN 2008) Daejeon, Korea.

[2] Burke T. OPC UA builds in security functions. www.automationworld.com/feature-2642

[3] Hannelius T. Integrating Industrial Information Systems with OPC UA – A Java reference Implementation- Master Thesis, Tampere University of Technology, 2009.

[4] Hannelius T. Salmenperä M., Kuikka S. (2008) Roadmap to adopting OPC UA. The IEEE International Conference on Industrial Informatics (INDIN 2008) Daejeon, Korea.

[5] Hannelius T., Shroff M., Tuominen P. Embedding OPC Unified Architecture

[6] IEC 61131-3 (2003) Programmable controllers- part 3: Programming languages, International Electro-technical Commission

[7] IEC/EN 61508 (2010). International Standard: 61508 Functional safety of electrical electronic programmable electronic safety-related systems Part1-Part7, Geneva

[8] Iwanitz F., Lange J. OPC, Grundlagen, Implementation und Anwendung. (OPC Fundamentals, Implementation and Applications) 2.Ed. Hüthig 2002

[9] Iwanitz F., Lange J., OPC fundamentals, implementation, and application. 3Ed. Heidelberg : Hüthig, 2006

[10] Kondor R. OPC Tunnelling Increases Data Availability, Matrikon, Inc. 2007

[11] Lange J., Burke T. J., Iwanitz F., OPC von Data Access bis Unified Architecture. 4 Ed. Berlin, VDE-Verl., 2010

[12] Lehnhoff S., Mahnke W., Rohjans S., Uslar M., (2011), IEC 61850 based OPC UA Communication – The Future of Smart Grid Automation. 17th Power Systems Computation Conference, Stockholm Sweden.

[13] Liu T. Cai G. Peng X., OPC Server Software Design in DCS. Proceedings of the 2009 4th International Conference on Computer Sciencne & Education 2009.

[14] Mahnke W., Leitner S.-H., Damm M. OPC Unified Architecture, Springer Verlag Berlin Heidelberg, 2009

[15] Maka A., Cupek R., Rosner J. OPC UA Object Oriented Model for Public Transportation System. UKSim 5th European Symposium on Computer Modelling and Simulation 2011.IEE Computer Society

[16] Michaud A. Creating Secure OPC Architectures, Matrikon, Inc. 2007

[17] Miyazawa I., Murakami M., Mutsukuma T., Fukushima K., Maruyama Y., Matsumoto M., Kawamoto J., Yamashita E., OPC UA Information Model, Data Exchange, safety and security for IEC 61131-3. SICE Annual Conference 2011, Waseda University Tokyo, Japan.

[18] Murphy E. OPC Security – Better Safe than Sorry. Matrikon Inc. 2006

[19] OPC Foundation, OPC AE 1.1 Specification, 2002

[20] OPC Foundation, OPC Batch Interface Specification, 2002

[21] OPC Foundation, OPC Commands 1.00 Specification, 2004 Draft

[22] OPC Foundation, OPC DA 3.00 Specification, 2003

[23] OPC Foundation, OPC DX 1.00 Specification, 2003

[24] OPC Foundation, OPC HDA 1.0 Specification, 2003

[25] OPC –Foundation OPC Overview OPC – foundation 1998

[26] OPC Foundation, OPC XMLDA 1.01 Specification, 2004

[27] OPC Foundation, OPC Unified Architecture Specification Part 2 Security Model, Version1.01 Feb. 2009

[28] OPC Foundation, OPC Unified Architecture Specification Part 6 Mappings, Version1.01 Feb. 2009

[29] PLCOpen, OPC UA Information Model for IEC 61131-3, http://www.plcopen.org/pages/tc4_communication/index.htm (Visited 09.2013)

[30] Reissenweber B. Feldbussysteme, 1998

[31] Renjie Huang, Feng Liu, Dongbo Pan, Research on OPC UA Security. 2010. 5th IEEE Conference on Industrial Electronics and Applications

[32] Sahin C., Bolat E. D. Development of remote control and monitoring of web-based distributed OPC system. Computer Standards & Interfaces 31, 2009, pp. 984-993

[33] Schwarz M.H., Börcösk J., Advances of OPC Client Server Architectures for Maintenance Strategies – a Research and Development Area not only for Industries. WSEAS Transactions on Systems and Control Issue3 vol. 3 2008. pp 195-207

[34] Son M. Yi M.-J. A Study on OPC Specifications: Perspective and Challenges. INFOST 2010 Proceedings. 2010

[35] Stopper M., Katalinic B. Servie-oriented Architecture Design Aspects of OPC UA for Industrial Applications. Proceedings of the international MultiConference of Computer Scientists 2009 Vo II IMECS 2009, Hong-Kong.

[36] Tan V. V., Yoo D.-S. and Yi M.-J. Design and Implementation of Web Services by using OPC XML-DA and OPC Complex Data for automation and Control Systems. (2006). Proceedings of the 6th IEEE International Conference on Computer and Information Technology. (CIT06)

[37] Tu, N. T. T., Thang H. Q. (2013) Design and Development of the Air Conditioning System by Using OPC UA Specifications and Modbus Protocol. IEEE 8th Conference on Industrial Electronics and Applications

[38] Yin Y., Zhou B. The Analysis and Research of OPC XML-DA Server. Energy Procedia 16(2012) (selection of 2012 International Conference on Future Energy, Environment, and Materials) pp. 1535-1540