A Survey on OLE for Process Control (OPC)

SCHWARZ M.H., BOERCSOEK J.
University of Kassel
Department of Computer-architecture and System-programming
Wilhelmshoeher Allee, 34121 Kassel
Germany

Abstract: - OLE for Process Control, also known as OPC, is a client-server architecture for exchanging process data. Although the first OPC-standard was published in 1996, and is today widely accepted and used in industries, it is still not very popular in academia, especially in Europe. The paper gives detailed information about OPC, and how OPC can be beneficial for research and development and gives an overview of the latest developments and standards.

Key-words: OLE for process control, OPC, client-server architecture, Unified Architecture

1 Introduction on OPC

It appears that about every two years an acronym from industries cleaves its way into academia; researcher and developer at universities start to show interest and then nothing happens. The publications about this particular topic are marginal and the interest vanishes. In 2002 the acronym of interest was "HIL" meaning 'Hardware in the Loop'. Still up to day a few companies are still using this acronym, mainly those who either invented it or were one of those who made it popular, but many research departments were using hardware in their development stages and development loops long before, without given it a name. So this was nothing original or new, but rather disappointing and at the end it was a name for something, which has been established already for many years.

When in 2005 the acronym OPC made its way into academia, again, many researchers were enthusiastic about OPC, but the interest decreased rapidly. The number of published papers is not increasing, especially in Europe. Although this acronym is again from industries, which makes a lot of researcher suspicious, but OPC is not a new and fancy name for something already used for many years, it can be a possibility for many universities, research units and institutes to get products on the market.

The roots and necessity of OPC has been set in the early 1980s, when networks and bus systems were developed, designed and established by companies and academia. Over [12] 50 different network systems were designed within few years for different applications. Few vanished into thin air, but most are used in a specific application area. And few, mostly developed by big companies, became standard, which were not always the best ones.

However, with the establishment of a number of different networks for different application or industry branches, the real problems started. For example CAN-Bus systems are used in cars, Profibus and its derivatives in process industries and most office networks are using Ethernet. For every single bus system, manufacturers have to develop drivers and maintain [3,11] them as shown in the figure below. In the past, companies who had developed specific bus systems, were not interested to let competitors into the market and did not give open access to the protocols, or published only parts of it.

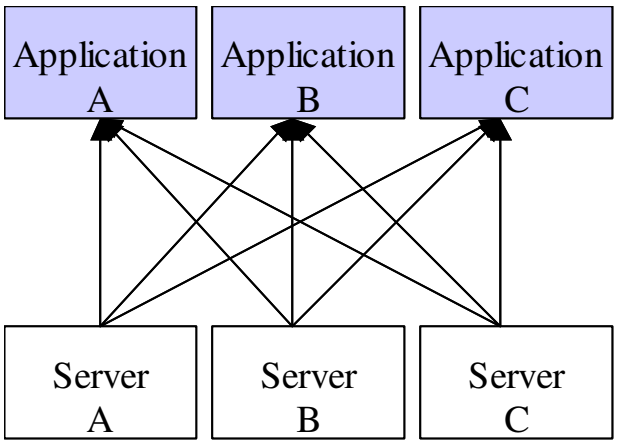

Fig. 1 : Communication with out OPC

If the manufacturers made changes on the bus systems, bus nodes or protocols, then system drivers had to be adjusted and re-tested, which is cost intensive and provides a high possibility of errors [3,11]. Additionally, costumers were not delighted if just because they bought few new bus nodes, parts of the bus system did not work or they could not use the new nodes [3,11,12]. Universities and institutes had basically no chance to sell their developments and get into the market even if the development was more innovating, and powerful; since already many medium sized companies had large difficulties and spent a lot of money to maintain their systems which was and still is impossible for universities.

In 1996, a few companies realised that the current situation was far from optimal. Therefore, a task force was established in the summer of 1995, with members of the companies: Fisher-Rosemount, Intellution, Intuitive Technology, Poto22, Rockwell [3] Software and Siemens AG to find a solution for the increasing problems.

Members from Microsoft provided support. This task force aimed to develop a standard based on Microsoft's (OLE/) DCOM technology, for the access of real time data under the operation systems Windows: which was named OLE for Process Control (OPC) [3]. It was ensured that an open participation was possible by incorporating feedback and obtaining acceptance from both industries and end-user. In December 1995, a draft version release was established for review by industries to provide feedback and to provide sample code. In August 1996 the first OPC specification was released. The figure below shows the OPC server / client approach.

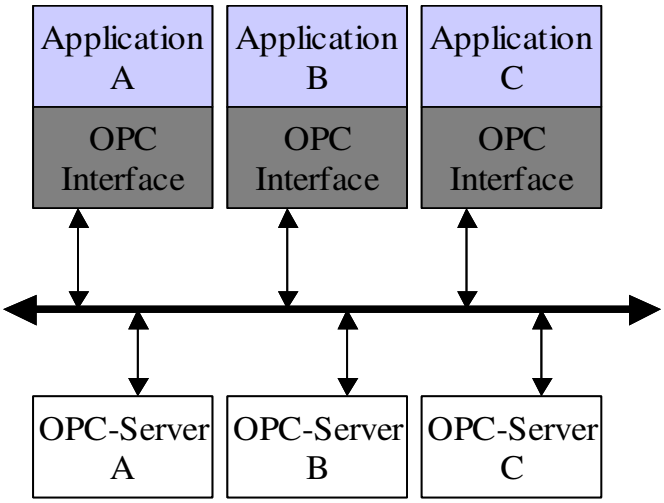

Fig. 2: Communication with OPC

Different and new specifications followed such as the OPC Alarms & Events specification, OPC Data Access Version 2.0 and Version 3.0, OPC Historical Data, OPC-Batch processes specification and so one. The remaining paper is structure as follows: A short review of the current available specifications is given in section 2. In section 3 current problems and solutions are presented, and a short introduction to OPC Unified Architectures and XML-web services is given. A possible future way of OPC is presented in section 4 and the paper closes with a short summary in section 5.

2 OPC

The OPC foundation released several specifications for different data communications on the bases of a client / server architecture. The OPC specifications are definitions of common interfaces to allow applications, OPC server, OPC client and devices to exchange data, events and information. The OPC driver and OPC interface need only to be implemented once. In the following the most common specifications are shortly detailed

2.1 Data Access (DA) specification

As stated in the introduction this specification was the first one released by the OPC foundation in 1996, currently release version 3.0 is the latest one. It defines an interface between a client and server to exchange process data [3,9,11]. The data access server allows one or several clients the connection to different data resources. It does not matter where the data resources are located, it could a data acquisition card on the same PC, sensors or control and automation units connected via a communication network. A data access client can also be connected to several data access servers. For further details it is referred to the OPC DA-standard [3,9,11].

2.2 Alarm and Event (AE) Specification

The Alarms and Events specification defines an interface for server and clients to transmit and acknowledge in a structured way occurred alarms and events. The AE-server can receive and capture data from different sources such as PLC (programmable logic controllers), control units and sensors, it can analyse data and decide if a an event occurred. It is important to note that AE server and DA server can have the same data sources [3,9,11]. The difference is that a DA-server provides a continuous data stream. The automated transition of values can be accommodated by a relative change of the value. This adjustment is only possible for analog values [11].. An AE server does not sent process values to a client, but the information that something happened or occurred, e.g. a valve has opened, or a temperature has reached its critical value. Criteria have to be defined and determined which are used by the server to decide that an event or alarm has happened. It is important to note that the specification does not oblige how the decision has to be executed or how a criterion has to be determined. An AE-server can directly receive the alarms or events from the process units or can receive the data from a e.g. DA-server [3,9,11].

2.3 Historical Data Access (HDA) Specification

The historical Data Access server provides a client with historical process data. It has to be distinguished between row process data and aggregated data, which is processed data. The aggregated data is created only on request from a client. The data access can be with the states readable, writable and changeable. Two different HAD-server client implementations exist [11]:

The first model structure offers simple trend data, which has only few optional interfaces implemented and the main duty of the server is to store row data. The second approach is a complex server with data compression and data analysis. The server can summarise data and analyse them, for instance it computes the mean value, minimal or maximal value etc. for the row data, and allows to renew data and to add comments. The specification does not state the sources of the historical data, which could be a database [11].. A HDA server is similar to a DA server,

but a HDA server does not have any objects such as OPC-group or OPC-Item. The client addresses directly data points via handles. The reason is that a DA-server provides a persistent access to process data, which are structured after certain criteria and therefore to insert or delete OPC-item objects are an exception. The number of process data a DA server provides are in the range of 100-1000 variables. The number of process variables a HDA-server supplies is in the range of 1000 to 10000. A client does not want to read this data persistently but maybe once a day or once week. Therefore, a different structure is used in comparison to DA-servers [3,9,11].

2.4 OPC Batch Specification

The OPC batch specification 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. Order data are sent and report information are received. Products for batch processing have to be manufactured according to the IEC 61512-1 [11]. 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, equipments 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 [3,9,11].

So far, the common used specification for OPC development are detailed. In the next section current problems and future directions of OPC will be described, analysed and evaluated.

3 Current Situation and Problems of OPC

New approaches such as XML web services and Microsoft's .Net technology are seen as new possibilities for industrial connectivity especially in connection with OPC and it is believed that they will replace Microsoft's COM / DCOM technology and its disadvantages [4]. This section will discuss this new approach and security issues when OPC is used in networks.

Additionally, a new specification OPC Unified Architecture (OPC-UA) is seen as a specification which will replace all the other previous OPC specifications such as OPC Data Access (DA) or OPC Alarms and Events (OPC-AE), which unifies all different OPC specifications [1,4], especially when using the new XML web services. Many managers and process engineers fear that the OPC-DA and OPC-AE server clients will be soon outdated and think about the point in time when they should swap the technologies [1,4,11].

3.1 COM / DCOM Technology

OPC communication is based on Microsoft's COM/DCOM technology. When OPC applications are installed on a PC then they using Microsoft's COM technology (Component Object Model) to exchange data. But when OPC applications are installed on two separate PCs then they using Microsoft's DCOM (Distributed Component Object Model). The COM messages are then basically wrapped in a Microsoft security layer, called DCOM [2,4,5]. Under certain circumstances, DCOM technology can detect timeouts which can lead to unreliable data transmission:

- Hardware problems, such as faulty network cards, router switches
- Overloaded networks
- Networks based on satellite links, wireless communication,…

Most networks have the same problems, an special action and preparations have to be taken [4].

An example will illustrate those problems. It is assumed that OPC-applications are running on two different PCs and are communicating and exchanging data via DCOM. After one application has sent a request, but before the second application has replied the communication link temporarily breaks [4]. The application can be forced to wait up to six minutes to recognise that an error occurred, even if the communication link is established again. It is not possible for users to change the six minutes time [4]. The demanded application just waits for DCOM to answer. All process data is during this time unavailable. And just imagine this application is a controller device which needs several data points per second to calculate the appropriate controller output to be send to the process. Software developers have to build a monitoring device around the DCOM communication to observe that it is functioning correctly [4]. XML web services is seen as the successor of DCOM, especially in combination with OPC unified architecture. But up to now, DCOM especially in combination with OPC-DA will continue, since DCOM is fast, which is necessary for real time requirements and applications. XML web services are at the moment still poor when it comes to applications which need fast process updates, such as field devices or monitoring systems [4].

Another important issue is that COM/DCOM is based on Microsoft operating system. Other such as UNIX, or Linux need special drivers and software to be used for OPC. However, many OPC-manufacturers and

developers provide such software and drivers so that OPC can be used on a non-Microsoft platform [2,3,4].

3.2 Security

Security is becoming more and more an important issue in process and automation industries. Until the last decade business network and process networks were strictly separated [5,6,8]. This was also a borderline for malicious software such as viruses and worms. This borderline will soften and probably will vanish within a few years time. Since the introduction of fieldbus systems and communication networks in industries, more and more devices are getting connected via such communication systems. PLCs can be easily programmed via a communication network, and engineers do not need to program each PLC manually by plugging a communication link, like a RS232 interface, into the PC or laptop and do the programming. Today, data and reports are already sent from the PLC to operator systems via Ethernet. Vendors and manufactures are planning to use Ethernet for the devices and sensors as well. Sensors, PLCs, PCs, operator systems will use Ethernet, but with different communication protocols. Process and IT engineers fear that all the problems with malicious network attacks will be carried into the process area and viruses and worms will corrupt the system, from PLC to the sensor area [5,6,8]. However, this scenario does already exist. Microsoft's DCOM supplies an easy to use communication framework for remote applications. DCOM allows software developer to reuse Microsoft's methods and functions in their own application. This was one of the reason why the OPC foundation selected DCOM as the base for OPC communication [5,6,8]. DCOM requests many ports for locating other hosts, resolving names, sending data. If these ports are unavailable, then DCOM starts automatically to search for others. All services and ports used by DCOM are targets for hackers. Has the virus infiltrated the system, then it has full access to all process components via the OPC-server. Therefore, the OPC-server is the largest risk factor but cannot be restricted in its methods [5,6,8]. Normally, the OPC-server gives full access to every client. However, the OPC-server has to be protected.

An easy but very restricting way to protect the server is to limit the traffic in only on way. Or to allow clients only access to certain tagged data, which can be set by the server with read/write or read only privileges. Additionally a special interface can be installed, which communicates with the OPC-server only via COM and provides the data and controls the privileges for clients using DCOM. This would separate the server from direct access by any client [5,6,8]. Another, and probably most promising way is to tunnel the data from the OPC-server to the OPC-client. Companies such as MatrikonOPC already provide such software. It works similar to a VPN connection. The advantage is that security features such as firewalls do not have to be sacrificed, but additionally DCOM can be remove from the systems [5,6,8]. A tighter security level and configuration can be used. The OPC-server accepts only data from the tunnelled connection from the OPC Client with the correct IP address. The data can be encrypted and looks from the outside just as a data stream. Standard TCP/IP, HTTP, HTTPS communications can be used and therefore DCOM is not necessary.

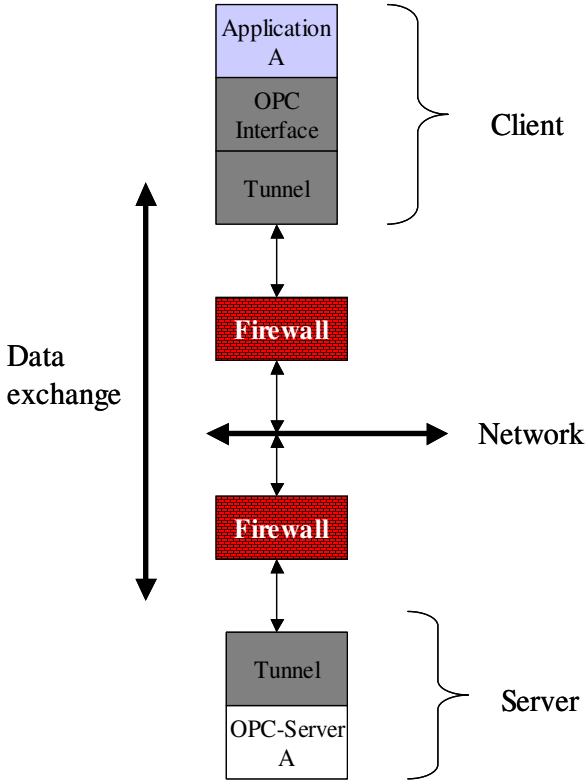

Fig. 3: Secure communication

This section briefly discussed security issues which arises using OPC, or other applications in communication networks.

3.3 Redundancy

In many industrial application for process and automation industries, redundancy is an important feature to increase the efficiency and reliability of the systems [6,7]. Redundancy is needed when either the communication link from the OPC server to the devices fails (Link based failure) or if the communication between the server and the client fails (Object based failures). Object based failures occur when the actual link between the client and server break down while link based failures occur when the physical link to the devices, control units or systems breaks. From this few point three different redundancy strategies exist and are listed below. Figure 4 shows all different redundant strategies.

- Device Level Redundancy
- Server Level Redundancy
- Application / client Level Redundancy

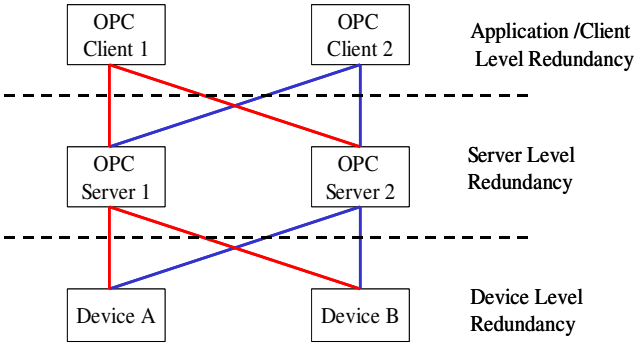

Fig. 4: Different redundant strategies

In a ***Device Level Redundancy*** strategy, controllers or data collection devices are implemented in a redundant configuration. Does the server-device connection fail, then the redundant device starts to operate. In this strategy no client is not involved. Vendors start to provide OPC server with redundant device support or redundant communication channel support [7,11].

Server Level Redundancy is available when two servers provide data to a client. One can be the primary server the other is in standby mode, or both are operating simultaneously. It is not compulsory that the client is designed for redundancy, since every client can be connected to several servers. If the communication between one server fails the redundant server supplies the necessary process data. It might be necessary to keep data on both servers consistent, especially if an OPC-server with historic data is implemented. Then a server has to possess an OPC client itself to update data from the second server as shown in Figure 5 [7,11].

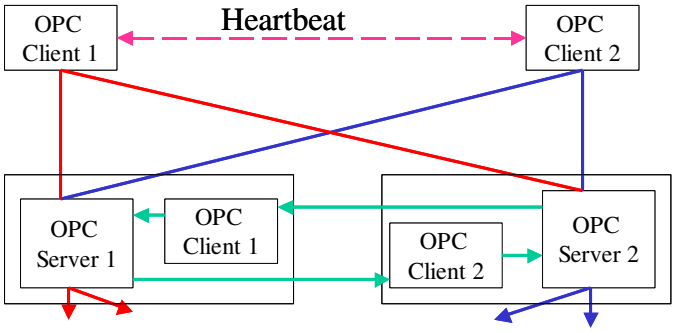

Fig. 5: Different possible connections when using redundant architectures

Application or Client Level Redundancy exists, when two clients or applications are implemented redundant. If the connection or link to a server or the application itself fails then the second client starts operating. On application or client level, at least a heartbeat signal has to be implemented to state that the current client is operating as shown in figure 5.More advanced structures to update and compare received data can additionally be implemented, again, based on a server client architectures [11].

Although, OPC server client architecture can be advanced with redundant architectures, it has no redundant strategies from base. Which means, additional and extra implementation and design is necessary to achieve redundancy, the original concept has not anticipated redundancy from the beginning.

3.4 XML-web services

XML web services are seen as the successor of COM / DCOM implementation and might become in process industries synonym for OPC connectivity and Microsoft's .Net technology [4]. XML web services are based on XML and very popular amongst different standards based bodies. It is easy to understand and is independent from a specific operating system. Developers are not tied to any programming language to implement web services. Applications developer can quickly create and use XML web services employing existing tools and framework. Web servers supply the essential infrastructure to exploit XML web services. Furthermore, this technology is accepted in practice, industries and in the business world. Figure 6 shows the general concept of XML web services [4].

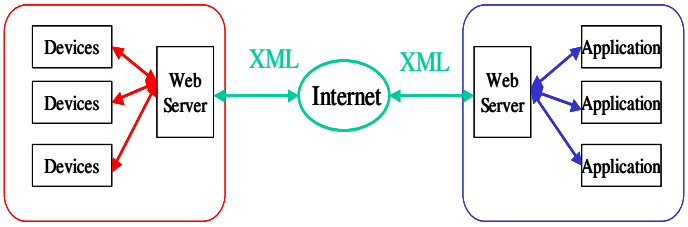

Fig. 6: Concept of XML web services

Currently, XML-web services still have few drawbacks. With XML-web services it is not possible to create a "report by exception" (RBX). It only can provide a "poll report by exception", which means poll once and the driver reports the changes while the services were disconnected. This method works well for data collection from remote sources but do not require real time information. So far this method is not applicable for real time application such as monitoring systems, or control devices. Another drawback is that XML messages are large in comparison with messages created by DCOM, which is not a problem for business applications but not suited for process real time applications [4].
However, XML-web services are a very new technology for applications in process and automation applications, therefore, this problems will be solved in the near future and the present limitations will vanish.

3.5 OPC Unified Architecture

Currently, OPC Unified Architecture [3,4,10] will not currently supersede OPC Data Access, OPC Alarms &

Events and OPC- Historical Data, but complement them. OPC-UA is based on XML-web services and is a platform independent standard. In general, the OPC-UA specification is organised in several specification chapters. Chapter 1 to 7 specifies the central potential of OPC UA. It defines the structure of the OPC address space and the services that are provided. Chapter 8 to 11 apply these capabilities to current specification of OPC-DA, OPC-AE and OPC-HAD [9,10]. The standard states how various systems, units and devices can communicate with each other by sending and receiving messages from server and clients via different types of communication networks. Also, it takes the lack of security from the previous standards into account. It defines a secure but robust communication which can identify and authorise clients and server and resists attack from harmful software. The new standard defines services which should be provided by servers for clients, and how servers can indicate which services they provide. Servers characterise the object models to be determined and investigated by clients. OPC-UA combines all three current standards and therefore clients can access process data, alarms and events data as well as historical data. Also, OPC-UA can be used with several communication protocols. The new standard intend a consistent, reliable and integrated address space and a consistent service model. An OPC UA server has the possibility to integrate different data types into one address space which can be accessed by clients using a defined and integrated set of methods [9,10].

The OPC-UA also includes redundancy concepts right from the beginning. Redundant clients and redundant servers can be designed and implemented in a consistent way, which can be used for a higher availability, higher fault tolerance or to balance the load a server or client is confronted with. Since a whole description of this standard would be far beyond of the scope of this papers, it is referred to the new standard [10].

4 Future Of OPC

Although, OPC United Architecture in combination with XML-web services has performance issue to be solved, such as the large size of XML messages and how it can be used in true real time applications, it will probably replace COM/DCOM technology, soon. Additionally, it provides an unified architecture from the bottom line of process control to the business line and applications like Computer Maintenance Management Systems (CMMS), Enterprise Resource Planning (ERP) and Enterprise Asset Management (EAM). Security and redundancy issues are considered in the standard right from the beginning. The current OPC standards and the new technology have still a high potential for practical research and development

5 Summary

Although, OPC is widely used in industries, it is hardly used in European academia. The matured standards have still enough potential to be used for research and practical applications. The new OPC-UA standard has many issues to be solved, which is also a good and interesting area for research. Since the new standard will be used from the bottom line (where still issues exist to be solved) to the top line of management, all different disciplines can work on different areas and are needed to understand all the different problems. Hopefully, European academia do not leave this area to a few companies, but getting involved.

References:

[1] Chisholm, A. DCOM, OPC and Performance Issues, *Intellution Inc*. 1998

[2] Harrison R.C. OPC DCOM White Paper, *Intellution Inc*. 1998

[3] Iwanitz F., Lange J. OPC Fundamentals, Implementation and Application, 2nd Rev.-Edition *Oldenburg*.2002

[4] Kondor R., OPC, XML, .NET and Real-Time Applications, *Matrikon, Inc*. 2007

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

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

[7] Murphy E. OPC Security – OPC Redundancy – Power of Prevention. *Matrikon Inc*. 2006

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

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

[10] OPC –Foundation OPC United Architecture, *OPC –foundation* 2006

[11] Pendli P.K., Gorbatchev V., Schwarz M.H, Börcsök J. 'OPC and its Strategies for Redundancy' International Control Conference (ICC2006-Control 2006), 30.08-1.9. *Glasgow, Scotland, United Kingdom*, ISBN 0 94769549, 2006

[12] Reissenweber, B. Feldbussysteme, *Oldenburg*.1998