DOI: 10.5604/20830157.1093182

OPC TECHNOLOGY AS AN AUTOMATED SYSTEMS INTEGRATION TOOL

Aibek Ashatovich Karatayev[1], Vasiliy Pavlovich Palshin[2], Zauresh Mukashevna Yarmuhamedova[1]
[1]K.I. Satpayev Kazakh National Technical University, Institute of Information and Communication Technologies,
[2]Kazakhstan & Germany University, Faculty of Engineering and Economic Science

Abstract. *The paper features the application of the OPC technology in automated systems. It gives an insight into the considered technology and provides the prospects of the OPC technology development.*

Keywords: OPC (OLE for Process Control), automated control system (ACS), component object model (COM), SCADA-system

TECHNOLOGIA OPC JAKO NARZĘDZIE INTEGRACJI SYSTEMÓW ZAUTOMATYZOWANYCH

Streszczenie. *W artykule rozpatrzono możliwość zastosowania technologii OPC w systemach zautomatyzowanych. Dokonano przeglądu tej technologii, a także określono perspektywy jej rozwoju.*

Słowa kluczowe: OPC, system sterowania automatycznego, Component Object Model (COM), system SCADA

Introduction

Software for the modern process automation systems is becoming more complex and costly. The development of the of software application sphere requires the use of more and more advanced tools and technologies. This is especially important in the development of complex software products that support different levels of Automated Control Systems (ACS).

A complex integrated system which covers automation at all business levels from the lowest control using sensors and actuating mechanisms to the level of company control mainly consisting of computing aids (personal computers, controllers, and other intelligent devices). Integration of automated system first of all means interaction between different software levels.

The ACS integration is aimed at real-time data exchange between different program systems developed using a variety of means that are installed on various platforms and operating on different computers. I.e. they should be aware how to request and send the required data to each other.

The technology was introduced by a consortium of world-famous hardware and software producers, such as the Rockwell Software, Fischer Rosemount along with Microsoft. OPC is a communication interface between different sources of data and software. This technology is based on the OLE/COM/DCOM architecture by Microsoft company [1].

1. Overview of OPC technology

OPC technology was developed to unify interaction of hardware and software mechanisms of Automated Control Systems. Within the frame of this technology the OPC-servers collect the data from controllers and transfer it to OPC-clients, for example SCADA-systems. Any OPC-client is able to exchange the data with any OPC-server independently of characteristics of equipment a definite OPC-server was developed for.

The main purpose of this technology is to provide software independence from definite hardware data sources for business dispatching systems developers.

The OPC-technology is meant to provide business software developers with a universal interface including a set of functions for data exchange between any appliances.

OPC was developed to ensure access of a client application to the lower level of a technological process in the most convenient form. Wide introduction of OPC technology in business has the following advantages:

- independence of applying dispatching systems from equipment used in a definite project;
- no need for modification of product software by its developers after equipment modification or manufacturing of new products;
- customer freedom to choose between equipment suppliers, as well as the possibility of integrating this equipment in the enterprise information system that covers the whole system of production control and logistics.

The basic idea of OPC-technology is that client software applications could receive data from the definite number of different sources, for example PLC, intelligent field equipment, data base management systems, other software, i.e. OPC is applied not only to data exchange with other hardware, but also to connection of one application to another [2].

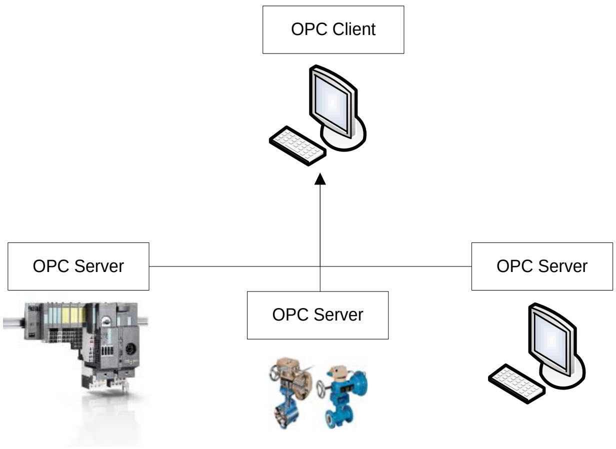

Fig. 1. Operation principle of Client – Server architecture

OPC technology is based on the Microsoft Distributed Component Object Model (DCOM), and sets requirements for the classes of data access objects and their specialized interfaces for use by developers of client and server applications. OPC technology as a means of interacting with a technical device can also be used to develop customized programs using C++, Visual Basic, VBA, or Delphi. The use of OPC in the development of customized software allows hiding from a developer the complexity of communication with the hardware, thus presenting a simple and convenient way to access the hardware via the COM object interfaces [5].

There are three main ways of obtaining data from an OPC server by an OPC client: synchronous reading, asynchronous reading and subscription. At synchronous reading a client sends a request to the server with a list of variables of interest to him and waits for the server to execute it. At asynchronous reading a client sends a request to the server and continues working. When the server has fulfilled the request the client is notified. In the case of subscription a client sends to the server a list of the variables of interest, and the server then sends to the client the information of the changed variables from his list on a regular basis. According to the OPC terminology these lists are called groups. Each client can simultaneously support many groups with different rates of updating.

The basic unit of data in OPC is a variable. A variable can be of any type allowed in OLE: various integer and real types, boolean, string type, date, currency, variant type, etc. Besides, a variable can be an array.

There is a fairly long list of OPC standards presented in Table 1. The OPC Foundation Consortium tries to cover all aspects of interaction with manufacturing equipment. Leading manufacturers of equipment and automation systems take part in the development of specifications; they try to take into account their own experience and provide someone who will use the OPC with absolutely everything he needs.

OPC specification describes two sets of interfaces:

- OPC COM Custom Interface;
- OPC OLE Automation Interface.

COM OPC Custom Interface group of standards describes the interfaces and procedures of OPC components and objects. This group of standards is designed primarily for high-level language software developers [5].

OPC OLE Automation Interface group is designed for software developers

Table 1. List of OPC standards

Name of standard	Designation
OPC Common Definitions and Interfaces	Common to all OPC Specification Interfaces
Data Access Custom Interface Standard	COM interface specification for operational data exchange
Data Access Automation Interface Standard	COM interface specification for operational data exchange, programming in Visual Basic languages
OPC Batch Custom Interface Specification	COM interface specification for hardware configuration
OPC Batch Automation Interface Specification	COM interface specification for hardware configuring, programming in Visual Basic languages
OPC Alarms and Events Custom Interface Specification	COM interface specification for event and alarm handling
OPC Alarm and Events Automation Interface Specification	COM interface specification for handling events and alarms, programming in Visual Basic languages
Historical Data Access Custom Interface Standard	COM interface specification for data storage
Historical Data Access Automation Interface Standard	COM interface specification for data storage, programming in Visual Basic languages
OPC Security Custom Interface	COM interface specification for processing data access rights

The applications that implement the abovementioned standards are called OPC servers. It is the OPC server that is responsible for receiving data from equipment, its processing and storage. The applications connected to OPC servers with the aim of receiving data from them are called OPC clients. An OPC client may connect to OPC servers supplied by one or several manufacturers. As follows from Table 1 the standards developed by OPC Foundation cover almost all the aspects of developing technological process automation control systems. In practice software developers implement only some of these specifications. The most popular are Data Access standard realizations [6].

2. Integration of OPC technology in automation systems

Figure 2 presents the diagram illustrating possible areas of OPC-server use in an enterprise automation system. There are several control levels:

- lower level – field buses and separate controllers;
- middle level – shop networks;
- technological process ACS level – SCADA type systems operation level;
- production ACS level – level of enterprise resources control applications.

Each of these levels could be serviced by an OPC-server supplying data to an OPC-client at a higher level.

An OPC DA server is the most widely used server in industrial automation. It provides data exchange, recording and reading between a client program and physical devices. The data consists of three fields: value, quality and timestamp. The data quality parameter allows communicating the information about the measured value exceeding the dynamic range, the lack of data, communication failure, etc. from a device to client software [3].

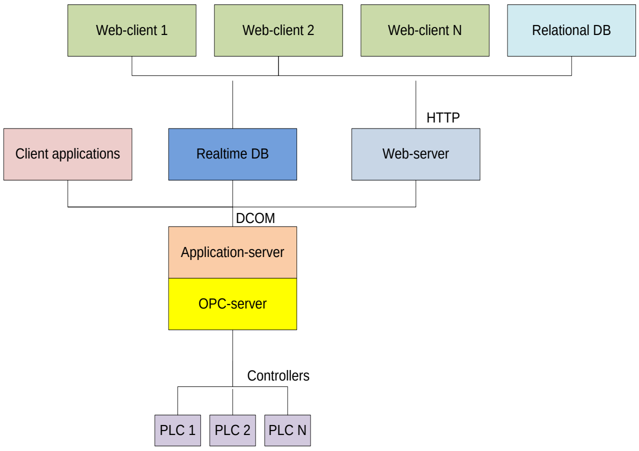

Fig. 2. Possible range of use of OPC-servers in enterprise automation systems

There are two standard modes of reading data from an OPC-server:

- synchronous mode: a client sends a request to a server and waits for reply;
- asynchronous mode: a client sends a request and switches to other tasks. After processing the request the server sends notification to the client, and the client receives the provided data.

In each of these modes data can be read either from the OPC-server cash, or directly from a physical device. Reading from cash is much faster, but the data can become outdated by the moment of reading. Thus a server should regularly update the data at the maximum allowed frequency. To reduce the processor load a frequency update parameter is used, so frequency may be set individually for each group of tags. Besides some tags may be made passive, then their values will not be updated by the data from a physical device [3, 7].

Recording data into a physical device can be performed by only two methods: synchronous or asynchronous; and it is recorded directly into a device without intermediate buffering. In the synchronous mode the recording function is active until a physical device provides confirmation of the record completion. This process could take a good deal of time during which a client is waiting for the completion of function and is not able to continue working. In the asynchronous record mode a client sends data to a server and goes on working. Upon completion of recording the server will send a corresponding notification to the client.

An OPC DA server may have a user interface which allows executing any auxiliary functions that simplify operating equipment. For example besides data exchange with SCADA an OPC server allows performing the following useful functions [7]:

- search of equipment connected to an industrial network;
- setting equipment parameters (designation, address, data exchange speed, watchdog timer period, control checksum availability, etc.);
- creating a hierarchical representation of tag names;
- observation of tag values;
- control of OPC server access rights.

In accordance with the standard an OPC server is automatically registered in the Windows register during installation.
The server is run similar to any other program or automatically from a client program.

In perspective open SCADA-programs an OPC interface could be included either as one of the interfaces for interaction with external programs, or as a basis structure of a SCADA-program. Tools for developing OPC-components could be supplied either by SCADA program developers or by independent software manufacturers. Using special software tools for development of OPC-servers and OPC-clients significantly simplifies the development of OPC-components, as it proposes ready OPC-interface implementation.

Examples of systems architecture including OPC-servers and OPC-clients are provided in Figure 3 and 4. A program in C++ language, for example SCADA-system, or a program in Visual Basic, VBA, or Delphi languages supporting the implementation of COM-objects (Figure 3) could be used as an OPC-client. A program in C++ language interacts with an OPC-server an OPC Custom interface, and a program in Visual Basic, VBA, Delphi languages – through an OPC Automation interface. OPC-servers and OPC-clients can work only on computers and controllers with operational systems supporting DCOM technology (for example, Windows XP and Windows CE) [1, 7].

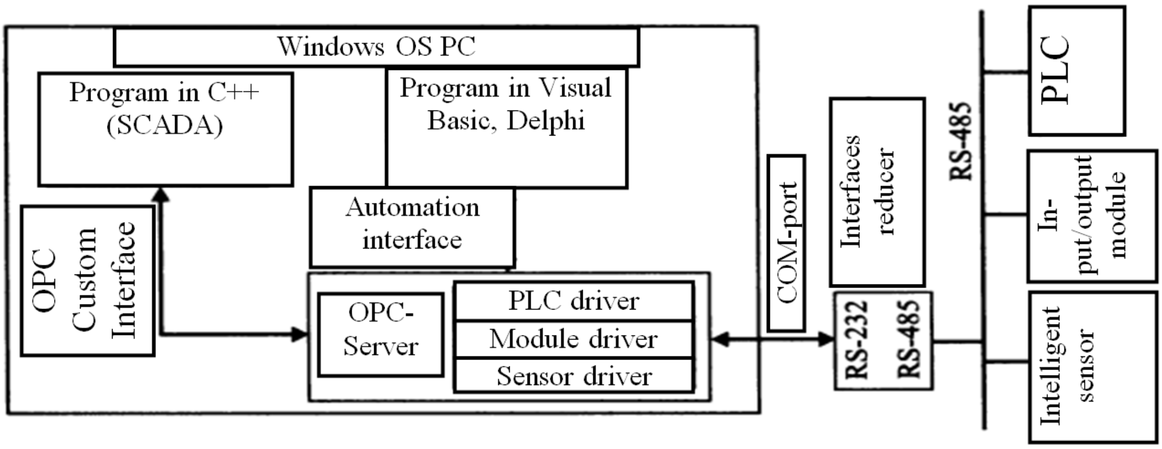

Fig. 3. Example of interaction of applications and physical devices through an OPC-server on one computer

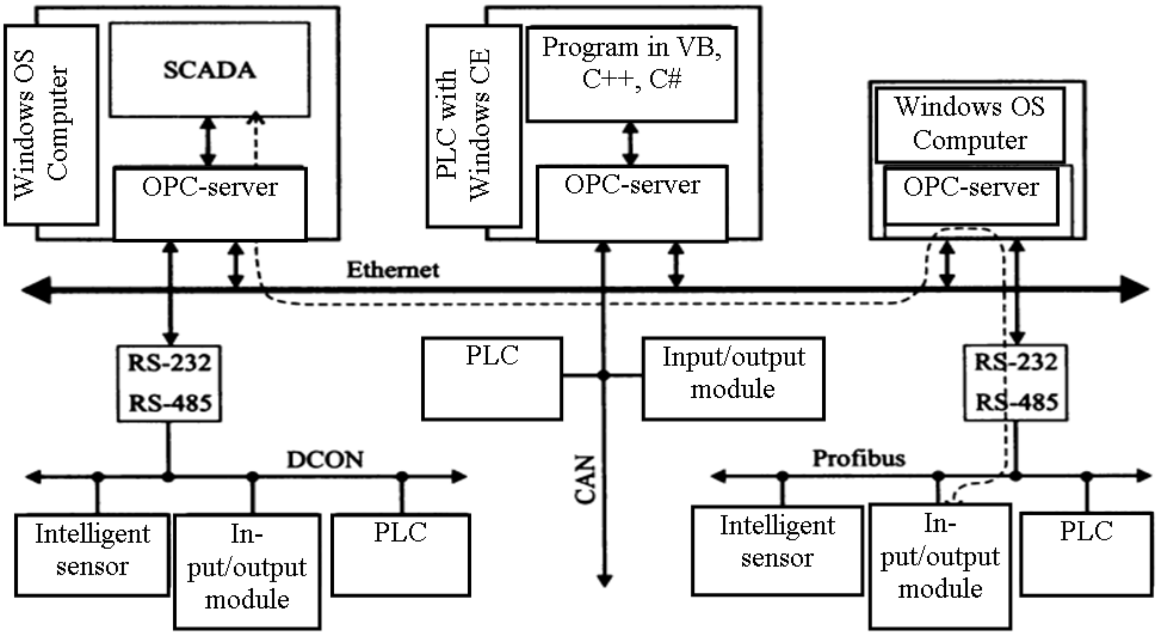

Fig. 4. Example of applying OPC-technology for network access to data in automation systems

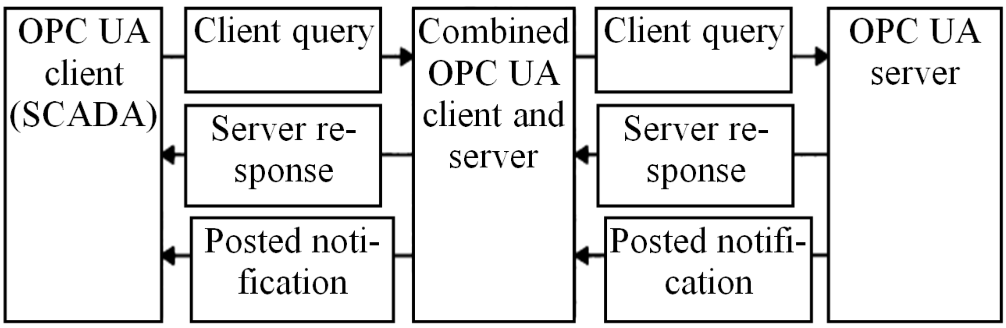

Fig. 5. Example of communication of OPC UA-clients and servers in one application

OPC-server connects to physical devices in many ways that are not provided in the standard. A client program and an OPC-server could be installed on the same computer (Figure 3) or on different computers of the Ethernet (Figure 4). In case there are several computers, each of them can have an OPC-server and physical devices connected to it.

In such a system any OPC-client from any computer is able to communicate with any OPC-server, including those installed in other computers of the network. This could be achieved due to DCOM technology which uses a remote procedure call (RPC). For example, a SCADA-system (Figure 4) may request the input/output module data via the path shown by a dashed line. It should be taken into consideration that computers and controllers in such architecture are able to operate with different industrial networks [1, 7].

Except the OPC DA technology there is a more advanced standard specification for data exchange in industrial automation systems which is called “OPC Unified Architecture”. It is considered a new generation of the OPC technology. The OPC UA sets message exchange methods between an OPC server and a client which do not depend on a hardware/software platform, as well as on the type of interacting networks and systems. A system based on the OPC UA may contain a lot of clients and servers. Each client can work with several servers simultaneously. Each server can serve several clients. User applications, for example a SCADA-system could create combined client and server groups for retranslating messages they are exchanging with other clients and servers as shown in Figure 5. An application software program, for example a SCADA system [3, 4] becomes a client during interaction with an OPC server.

3. Conclusion

As of today OPC is a perspective technology for integration of hardware and software in automation systems. The OPC proposes standards for exchange of process data with substantial opportunities. But there is still a lot of equipment and software which are not covered by OPC-technologies. However, Microsoft Corporation does not provide COM/DCOM anymore, these technologies have been replaced by more advanced, for example NET. Yet only OPC DA standard is widely used. Nowadays a lot of manufacturers equip their products with OPC DA servers. The OPC HDA standard has been effectively developed recently, still other specifications have not made such progress. The more advanced OPC UA technology is a promising direction.

OPC technology is a powerful and unified tool. It has made substantial contribution to standardization of embedded systems interacting with computers. A lot of modern developers successfully apply it in the development of technical systems.

References

[1] Denisenko V.V.: Computer control of technological processes and experiment. M.: Hot line, Telecom, 2009.

[2] Fedorenko D.: OPC–client programming in C++ and C#: Manual, 2011.

[3] Kozletsov A.: OPC UA – integration standard for production information systems, materials of the third international scientific and practical conference "Efficient production management technologies" (MESA International), October 7-18, Moscow, 2011.

[4] Levin S.: OPC technology in safety systems, Protection technologies Journal, No. 5, 2011.

[5] Pyavchenko T.A.: Automatic control system of technological process designing in SCADA-system. Manual in technical disciplines "Automated information and control systems" and "Integrated design and control systems", 2007.

[6] Shishov O.V.: Modern industrial automation technologies: manual, Saransk: Publishing house of Mordva University, 2007.

[7] Official site of OPC standard: www.opcfoundation.org

Aibek Ashatovich Karatayev
e-mail: aibek.karatayev@gmail.com

Graduate student of Institute of Information and Telecommunication Technologies of KazNTU named after K.I. Satpayev. Has experience in the field of information technology, particularly in software development. Diploma work related to software development of the simulation model of motion of small space systems around the center of mass..

Ph.D. Vasiliy Pavlovich Palshin
e-mail: aibek.karatayev@gmail.com

Vasiliy Pavlovich Palshin was born in 1945. In 1968 he graduated from the Kazakh Polytechnic Institute. In 1979 he defended his PhD thesis. For the current moment he is a Docent of Kazakh German University. The main scientific field includes the works in such spheres as: automated control systems, identification and mathematical modeling. More than 60 scientific papers were published by him.

Prof. Zauresh Mukashevna Yarmuhamedova
e-mail: aibek.karatayev@gmail.com

Zauresh Mukashevna Yarmukhamedova was born in 1949. In 1971 she graduated from the Kazakh Polytechnic Institute. In 1979 she defended her PhD thesis. For the current moment she is a Professor of Kazakh National Technical University and member of International Informatization Academy. The main scientific field includes the works in such spheres as: mathematical modeling of process facilities and optimized control. She is the author of more than 80 scientific papers.

otrzymano/received: 28.09.2013 *przyjęto do druku/accepted: 12.02.2014*