Roman KWIECIEŃ

ORCID: 0009-0003-9058-6497

DOI: 10.15199/48.2025.05.01

Analysis of the functionality of OPC DA technology in the Windows development process

Analiza funkcjonalności technologii OPC DA w procesie rozwoju systemu Windows

***Abstract**. OPC technology is a communication standard in computer-based industrial automation systems. Its functionality allows the transfer of information between computer applications operating in the client-server scheme. The original concept of this technology, called Classic, uses Windows' Component Object Model (COM) technology and depends on updates to this operating environment. This article presents the impact of a single modification of the COM technology on the operation of the communication module working as a client of the OPC technology.*

***Streszczenie.** Technologia OPC jest standardem komunikacyjnym w komputerowych systemach automatyki przemysłowej. Jej funkcjonalność pozwala na przekazywanie informacji pomiędzy aplikacjami komputerowymi pracującymi w schemacie klient-serwer. Pierwotna koncepcja tej technologii, nazwanej jako Classic, wykorzystuje technologię COM (ang. Component Object Model) systemu Windows i jest zależna od aktualizacji tego środowiska operacyjnego. W niniejszym artykule przedstawiono wpływ pojedynczej modyfikacji technologii COM na działanie modułu komunikacyjnego pracującego jako klient technologii OPC.*

Keywords: OPC Technology, OPC Data Access, Advise time, Communication Driver, SCADA/HMI.
Słowa kluczowe: Technologia OPC, OPC Data Access, Czas Advise, Sterownik komunikacyjny, SCADA/HMI.

Introduction

In industrial automation systems, there is a strong tendency to create distributed computer systems. This systems are becoming increasingly important in the information exchange process. For the needs of such systems, IT systems for technological process management are implemented. Among them, the following can be distinguished (Fig. 1):

- Enterprise Resource Planning (ERP),
- Manufacturing Execution Systems (MES),
- Supervisory Control and Data Acquisition (SCADA).

Enterprise Resource Planning (ERP) is an integration of an IT system used to manage enterprise resources or to facilitate the cooperation of a group of cooperating enterprises by collecting data and enabling operations on the collected data. ERP software includes a number of modules (e.g. production, orders, billing, data warehouse, transport, human resources, etc.) that can be integrated with the company's software. In developing the ERP II system class, Internet technologies and the XML language standard were used. This allows for full interaction of the system with the environment by exchanging data via computer networks. As a result, this interaction contributes to the optimization of business processes in the enterprise-partner relations.

Manufacturing Execution System (MES) is a computer system that uses information technology, software, electronic devices and automation components that allow for the collection of information from production stations that allows for the optimization of production process operations in the business area. A typical MES system performs the following functions:

- Process Management,
- Performance Analysis,
- Production Tracking and Genealogy,
- Quality Management,
- Data Collection and Acquisition,
- Document Control,
- Resource Allocation and Status,
- Labour Management,
- Dispatching Production Units,
- Operations / Detailed Scheduling,
- Maintenance Management.

The system supervising the course of a technological or production process SCADA/HMI (Supervisory Control and Data Acquisition/Human Machine Interface) is a computer system whose main functions concern the collection of current data from the process, visualization of its status, superior control, alarming and recording events, data archiving and sharing information about the process in computer networks. The selection and amount of presented process data corresponds to a specific category of the supervised process and current service requirements. The process data is visualized on synoptic screens, which display values in the form of numbers, or using various types of charts, sliders or meters. The construction of synoptic screens is performed in all computer applications intended for the creation of SCADA systems, which contain libraries of ready-made graphical presentation elements.

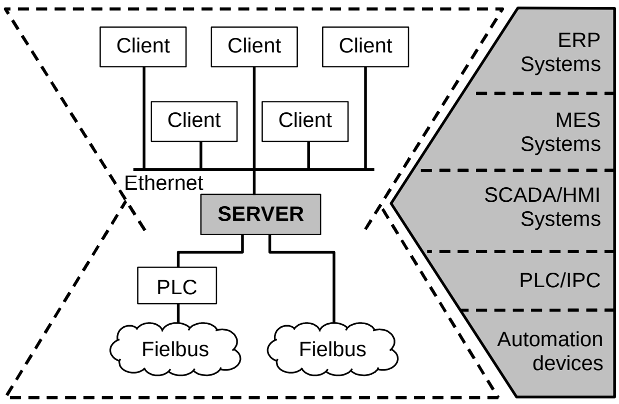

Fig.1. A model of IT systems infrastructure reflecting the technological process [1, 2]

At the lower levels of the technological process are used industrial control systems CNC (Computer Numerical Control), PLC (Programmable Logic Controller), IPC (Industrial PC), sensors, actuators and others industrial automation devices. The basic task of this system is to control devices in order to carry out the tasks of the technological process and to collect information on the basis of which the IT system will be properly managed in the decision-making process [1].

The industrial automation IT system can be divided into two subsystems, in the form of a model consisting of two pyramids (Fig. 1) [1]:

- production subsystem (lower pyramid), in which the master unit (server) reads process data from the technological process (from industrial automation devices) and collects them in the industrial database. It becomes part of the information describing the functionality of the enterprise.
- management subsystem (upper pyramid, reversed to the lower one), in which the main role is played by supervisory units from the highest layers of production management. Based on the data collected in the industrial database and their technological and economic analysis, decisions are made to find an appropriate reflection of the functioning of the systems controlling the technological process.

The key element of the entire IT system are servers, which are a link in the process of exchanging information between two subsystems. The servers make process data available to clients according to a strictly defined communication protocol in the Ethernet network. The currently common standard for information exchange in the client-server scheme of industrial automation devices is the OPC technology (OLE for Process Control).

OPC Technology

In 1995 the most developed IT technology in Windows was OLE (Object Linking and Embedding). It was used to exchange data between computer applications. Industrial automation IT systems lacked a communication standard that would allow remote management of the technological process. For this reason, the OPC Task Force was formed, which included the following companies: Fisher-Rosemount, Intellution, Intuitive Technology, Opto22, Rockwell and Siemens AG. The goal of this group was to develop a communication standard between industrial devices. This work resulted in the publication of the first OPC (OLE for Process Control) specification in August 1996, and in September of the same year the “OPC Foundation" was established. Further coordinated efforts sought to maintain and publish new OPC specifications. The name of the specification implementing specific COM/DCOM interfaces is identified with the name of the corresponding OPC server type. The specifications define separate tasks for OPC servers in terms of their functionality and include [2, 3]:

- OPC Data Access (OPC DA) – allows access to current process data online,
- OPC Historical Data Access (OPC HDA) – allows access to archived data,
- OPC Alarms & Events (OPC A&E) – broadcasts events in the system and reported alarms,
- OPC Security – defines the method of access to data,
- OPC Batch – is required when managing batches,
- OPC XML-DA – integrates OPC technology and the XML markup language (eXtensible Markup Language) for working on the Internet [4].
- OPC Unified Architecture (OPC UA) – enables communication at the level of various operating systems. It is a platform-independent standard in which messages exchanged between clients and
- servers can be sent using various types of networks. OPC UA was created to ensure that data is exchanged efficiently, securely, resistant to attacks and ensuring mutual identification of the client and server. Additionally, it defines a rich set of services through which customers can obtain data [5-10].

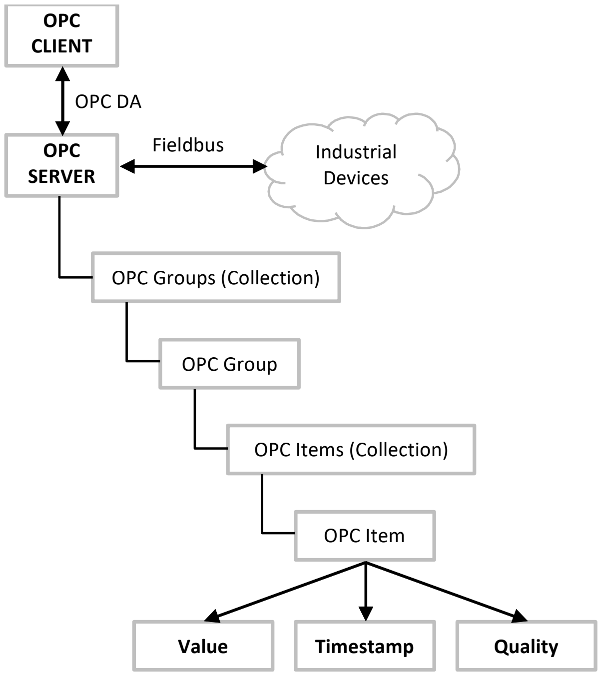

Fig.2. Architecture of access to process variables via OPC DA [1]

The OPC DA specification allows access to a single process variable (OPC Item) with the ability to read or write, each of which has a Value, Timestamp, Type and Quality, as shown in Figure 2. The timestamp can be generated by the OPC DA client or by the OPC DA server if the client does not have this capability. Using this specification, it is possible to only view the values of current process variables or change its value. Due to the complexity of the processes carried out by the OPC DA server, process variables have been logically divided into groups (OPC Group). In these groups, the variables are characterized by different scanning times and reading modes.

Models of access to process data

According to the OPC Data Access specification, an OPC DA server consists of the following objects [1]:

- OPCServer server object – provides a number of properties and methods and one ServerShutDown event that informs the client about its closure,
- OPCBrowser object – provides methods and properties that allow the client to browse the namespace,
- OPCGroups object – includes a list of OPCGroup group objects,
- OPCGroup object – offers all methods, including those available in the custom interfaces of the OPCGroup object,
- OPCItems object – consists of a list of OPCItem type objects and is used to manage them, e.g. adding, removing, etc.,
- OPCItem object (Table 1) – is the lowest level in the automation object hierarchy, representing a process variable; the client is able to read and write the object value using appropriate methods; this object does not have events; creating a process variable requires first of all providing its full name, activity state and specifying the data type..

OPC DA technology interfaces specify a set of functions that allow mainly for initializing process variables, writing and reading their values from the OPC DA server (tab. 1). The initialization of these variables can take place with the activity status True or False. Inactivity means that the variable exists in the OPC DA server collection, but its value is excluded from the update cycle and is not broadcast to the OPC DA client. For this reason, two types of communication models arise between the server and the OPC DA client:

- Add Inactive Tags – AIT (Fig. 3),
- Add Active Tags – AAT (Fig. 4).

Tab.1. OPCItem object definition

Properties	Methods
Parent ClientHandle ServerHandle AccessPath AccessRights ItemID IsActive RequestedDataType Value Quality TimeStamp CanonicalDataType EUType EUInfo	Read Write

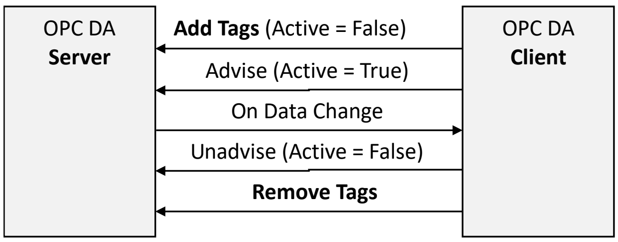

Fig.3. The process of advising tags on the OPC DA server – AIT model

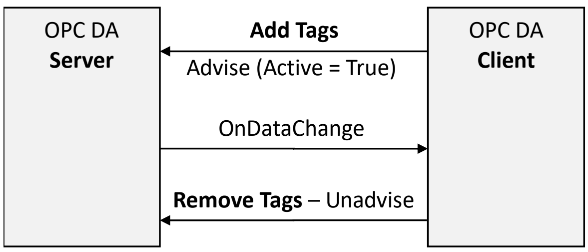

Fig.4. The process of direct advising tags on the OPC DA server – AAT model

In the first AIT model (Fig. 3), during initialization, the OPC DA client adds all process variables on the server with their inactivity. When the OPC client requests data values using the Advise function, the required process variables are activated. From this moment, the OPC DA server sends the variable values to the OPC DA client in asynchronous mode (OnDataChange), i.e. when the value of this variable changes. The entire process continues until the Unadvise function is called to deactivate the variable. Before the end of the client's work, the set of process variables is finalized and deleted from the OPC DA server.

Initialization in the second AAT communication model (Fig. 4) does not add all process variables to the OPC DA server. The addition process takes place when the client requests the required set of process variables with an active status using the Advise function. Then the server sends the current variable values to the client and those that have changed. During a work, it is possible add and remove variables to retrieve their values and input new data to the server. Ending the client also removes all variables from the OPC DA server if they were not previously eliminated with the Unadvise function.

Analysis of activation times of process variables

The access time to the value of a process variable is a leading parameter in a computer system of industrial automation, especially when their number reaches the level of several tens of thousands or even hundreds of thousands. In most cases, the largest set of process variables is consumed by the alarm server. There are cases where approximately 20 000, 60 000 or more variables are used for alarming. Obtaining data values from technological lines depends on the load of the local computer network. Depending on the communication technology used with the process device, the data reading speed can be up to approx. 30 000 variables values per second. Therefore, it is expected that the activation time of variables on the OPC DA server is as fast as possible. Lists of reading times for variable values are shown in Tables 2 and 3 for two communication models in the client-server scheme.

Tab. 2. Times of direct advising 1000 tags – AAT model

No	Advise time
1	281 ms
2	219 ms
3	187 ms
4	172 ms
5	172 ms

Tab. 3. Advising and initialization times for 1000 tags – AIT model

No	Initialization time	Advise time
1	187 ms	31 ms
2	140 ms	16 ms
3	141 ms	31 ms
4	125 ms	31 ms
5	141 ms	15 ms

The data from Table 2 show that when using the AAT model with direct activation of variables on the OPC DA server (Fig. 4), Advise times are even approximately 10 times higher than when using the AIT model with no activation (Fig. 3 and Table 3). Only in the case of initialization, a longer time for adding process variables to the server is recorded. However, this only happens once on start the OPC DA client.

The analysis of Advise times for two communication models of the client with the OPC DA server shows that the model with no activation during AIT initialization is a more optimal solution because it has much lower Advise times. Software vendors use this model to provide faster process image refresh times. This is particularly important when there are a large number of monitored values of process variables.

Windows and OPC DA technology

Since 2023, incomplete operation of OPC DA clients using the AIT model with no activation of process variables has been noticed. After analysing the operation of the OPC DA client, there is a loss of transmission of variable values after their activation on the OPC DA server. Only those variable values that have changed on the server were broadcasted. A situation occurs when the OPC DA client activates process variables, while the computer system registers some of these variables as inactive in the system.

The visualization system does not show the required information and the alarm server does not have access to current entries. The detection of this fault resulted in modification of the functionality of the OPC DA client module. Its correct operation is ensured by additionally forcing the variable values to be read from the OPC DA server after its activation (AIT + Read). For this purpose, the ReadItems method was additionally defined for the

OPCItems object. This operation increased the Advise times of the variables by approximately 30%, as shown in Table 4. In the coding of the Advise function, an extra line appeared in the script part, as shown in Listing 1.

Tab. 4. Advising and initialization times for 1000 tags

No	Initialization time	Advise time
1	234 ms	47 ms
2	141 ms	31 ms
3	125 ms	47 ms
4	125 ms	47 ms
5	141 ms	32 ms

Listing 1. Example of the process variable Advise function

```
List : TdOPCItemList
…
for i = 0 to 9 do
 List.Add( … );
…
OPCStdGroup.OPCItems.SetActive(List, True);
OPCStdGroup.OPCItems.ReadItems(List);
```

Conclusions

OPC Classic technology based on COM/DCOM Windows technology is the first standard introduced into industrial automation systems. It made building extensive control and measurement systems easier and allowed for integration with other computer systems. Despite the introduction of the new OPC UA (Unified Architecture) technology, OPC Classic technology is currently still used in factories around the world. The cooperation of the author of this article with a German company dealing with the automation of industrial processes in car paint shops confirms the use of RSLinx OPC Server and FactoryTalk OPC Gateway servers for communication with Rockwell PLC controllers.

Communication modules operating as OPC DA clients adapted to the Windows environment are installed in the SCADA/HMI computer system. The fact that this system was updated resulted in incorrect broadcasting of process variable values to OPC DA clients. The author of the article removed a defect in the functionality of this communication module in mid-2023 at a car paint shop in Alabama. For this reason, the aim of the article was to present two communication models between the client and the OPC DA server to illustrate the correctness of choosing the "AIT + Read" model used in the communication module of the SCADA/HMI system (Fig. 5). The solution to the communication problem and its effects were also presented, which are shown in table 4, thus illustrating approximately 30% longer execution times of the Advise function.

The execution of the process variable advisory function is very important for SCADA/HMI systems. During the visualization, the process image view changes, which is associated with the continuous activation and deactivation of tags. The system must keep up with these changes so that the communication drivers exclude from their circulation the reading of variable values from PLC controllers. Sometimes tags are left active all the time, so that the viewing of the process image is unnoticeable to the operator. The refreshing of visualization images can also be accelerated by remembering the last values of tags, to show ready values, and after the reading cycle time, display the current ones.

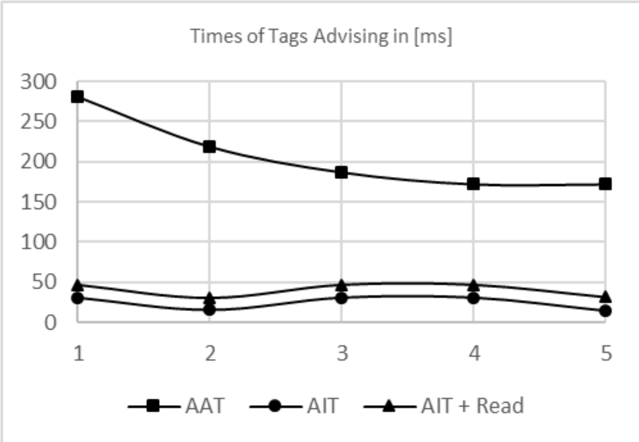

Fig. 5. Summary result of advising times for three cases: classic adding tags (Active = True) – AAT model; Advise (Active = True) after adding tags without activation; Advise + Read Values (Active = True) after adding tags without activation

***Authors**: dr inż. Roman Kwiecień, Uniwersytet Radomski, Wydział Transportu, Elektrotechniki i Informatyki, ul. Malczewskiego 29, 26-600 Radom, E-mail: r.kwiecien@uthrad.pl.*

REFERENCES

[1] Kwiecień R.: Komputerowe systemy automatyki przemysłowej, Computer systems for industrial automation, ISBN: 9788324651429/978-83-246-5142-9, Pub. Helion, Gliwice (2013).

[2] Kwiecień R.: Technologia OPC jednorodnym medium w procesie dostarczania informacji w logistycznym systemie komputerowym przedsiębiorstwa, OPC technology as homogeneous medium in the process of providing information in the logistics enterprise computing system, LogiTrans, Szczyrk, 7-10 IV (2014), ISBN 978-83-7351-424-9, Logistyka 3/2014, ISSN 1231-5478

[3] Iwanitz F., Lange J.: OLE for Process Control. Fundamentals, Implementation and Applications, Huthig Verlag heiderberg, RFN, (2001).

[4] Michalski A., Makowski Ł.: Praktyczne użycie XML w rozproszonych systemach pomiarowo-sterujących; Przegląd Elektrotechniczny 05/(2008) Str. 255-258.

[5] Pu Ch., Ding X., Wang P., Xie S., Chen J.: Semantic Interconnection Scheme for Industrial Wireless Sensor Networks and Industrial Internet with OPC UA Pub/Sub; (2022) https://doi.org/10.3390/s22207762

[6] Ioana A., Korodi S., Korodi A.: Improving OPC UA Publish-Subscribe Mechanism over UDP with Synchronization Algorithm and Multithreading Broker Application; (2020) https://www.mdpi.com/1424-8220/20/19/5591

[7] Shin I., Song B., Eom D.: Auto-Mapping and Configuration Method of IEC 61850 Information Model Based on OPC UA; (1017) https://doi.org/10.3390/en9110901

[8] González I., Calderón A. J., Figueiredo J., Sousa J. M. C.: A Literature Survey on Open Platform Communications (OPC) Applied to Advanced Industrial Environments; (2019) https://doi.org/10.3390/electronics8050510

[9] Gutierrez-Guerrero J. M., Scholar and Juan Antonio Holgado-Terriza: Automatic Configuration of OPC UA for Industrial Internet of Things Environments; (2019) https://doi.org/10.3390/electronics8060600

[10] Arestova A., Martin M., Hielscher J., German R.: A Service-Oriented Real-Time Communication Scheme for AUTOSAR Adaptive Using OPC UA and Time-Sensitive Networking; (2021) https://doi.org/10.3390/s21072337