# New trend on OPC Middleware

Simona-Anda GHERASIM, Adrian Mihai GĂITAN, Vasile Gheorghiță GĂITAN, Valentin POPA
*Stefan cel Mare University of Suceava*
*str.Universitatii nr.13, RO-720229 Suceava*
sgherasim@stud.usv.ro, adgaitan@stud.usv.ro, gaitan@eed.usv.ro, valentin@eed.usv.ro

***Abstract* — A new trend on OPC Middleware is presented in this paper. In addition to general solutions of middleware (CORBA, Java RMI, DCOM), the attention focuses over OPC Foundation specifications, meaning: OPC Data Access, OPC Alarms & Events, OPC Historical Data Access, OPC XML-Data Access, OPC Unified Architecture and OPC Express Interface. OPC Unified Architecture (OPC UA) signifies a new standard of the OPC Foundation, providing interoperability in process automation and beyond.**

**Time specific to read OPC items on Windows and Linux platforms is also emphasized. At the end of this paper, strategies of migrating to OPC UA applications are introduced.**

***Index Terms* — Middleware, Object Linking and Embedding for Process Control (OPC), OPC Data Access, OPC XML DA, OPC Unified Architecture**

## I. INTRODUCTION

The concept of "**Middleware**" has been used within broad thoughts. It generally refers to all intermediary software levels, which supports communications carried out between client and server. The concept can also unify or interpose between two disjointed applications, or between components of the same application. The solution relies on the Internet network, where processes interact on one or more devices; the interaction occurs due to the middleware applications. The databases, web servers or the content management systems are some of those systems using such applications. A common middleware application allows the using of a written program, so as to access a database; on its turn, this will allow the access to another database.

The middleware programs provide services of messages, so that different applications can communicate. All these different applications are reunited by means of middleware, into a system of integrating the applications, assigned as "the enterprise application integration" (EAI).

The general solutions of **Middleware** are the following: **CORBA (**Common **O**bject **R**equest **B**roker **A**rchitecture**), Java RMI (Java R**emote **M**ethod **I**nvocation**) and DCOM (D**istributed **C**omponent **O**bject **M**odel**)**. But, these conditions are not enough. They should solve specific problems in a standard way. Therefore, the need of standardization at device's levels should be satisfied. Two new solutions can be emphasized: "**Data Acquisition**" and "**OPC Data Access**".

## II. OBJECT LINKING AND EMBEDDING FOR PROCESS CONTROL (OPC). WHAT DOES THE OPC MEAN?

The **Middleware** is quite a wide field, which has been continuously progressing. Amongst the Middleware applications, one might mention: **OPC**, **ACE, IBM, Oracle** and **SUN.**

The OPC (OLE for Process Control) has lately become a well-known standard, specific to communications between SCADA systems and the field devices. The reason consisted in reducing the number of drivers, necessary to operating with more field devices.

Since its very beginning, the OPC standard has relied on DCOM architecture, defined by Microsoft. Few implementations rely upon other already existing operating systems.

Though, OPC has been also implemented to UNIX. The reasoning of such implementation depicts that OPC is not the only protocol, specific to all communication oriented devices. The possibilities of this implementation are: DCOM on Linux, OPC XML Data Access and OPC Unified Architecture.

More tests were performed, so as to check if the OPC sever on Windows can also be ported on Linux; the time necessary to read the OPC elements on both platforms was also taken into account. After the tests, proofs depicted that OPC on Windows read about 100,000 elements per second, while OPC on Linux read only 76,000 elements per second.

As concerns OPC, the server is shown as a list of elements, whose values can be read or modified. The existing interfaces relate to searching the servers and to discovering their name spaces. The clients can individually organize their access to the server of elements, by their grouping using OPC Group. [1][2][6][17]
Fig. 1 illustrates the architectures with or without OPC.

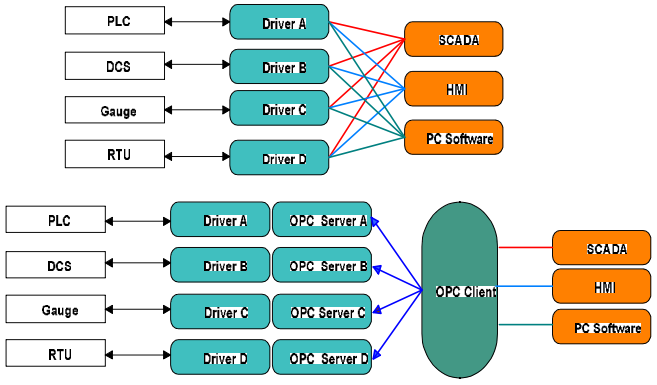


**Figure 1.** Architectures with or without OPC [19].

Nowadays, Microsoft facilitates other distributed architectures besides DCOM, meaning the web services based on XML and W3C.

OPC Foundation has launched a first version related to specification conditions, needed to "a unified architecture"; this proved to be of complete interoperability. The scalability is also unlimited, looking beyond the platforms and of fully Internet connection.

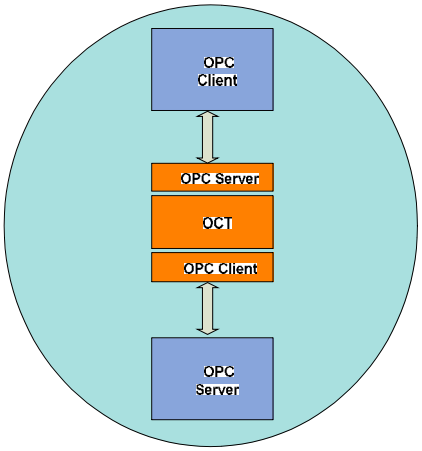


**Figure 2.** The client-server model of Middleware OPC [18].

**OPC** middleware products are conceived so as to develop any application based on OPC. A set of intelligent OPC products will simplify the operating and commissioning carried out by the OPC clients and servers, will significantly improve the data transferred within OPC interfaces, and will also ensure reliable communication connections.

OPC was initially defined as a standard solution to the recurrent tasks of the PC – applications connection (for instance, HMI (Human Machine Interface) or SCADA systems), with devices of automation and process control. Nowadays, the OPC standard has progressed at the level of a robust operator of data transferring, able to accomplish the entire planning document, specific to resources of even video signals. [1][2][6][17]

## III. OPC SPECIFICATIONS

For fourteen years, the OPC Foundation has launched different specifications (specification conditions) within Middleware, starting with OPC Data Access and continuing with OPC Unified Architecture.

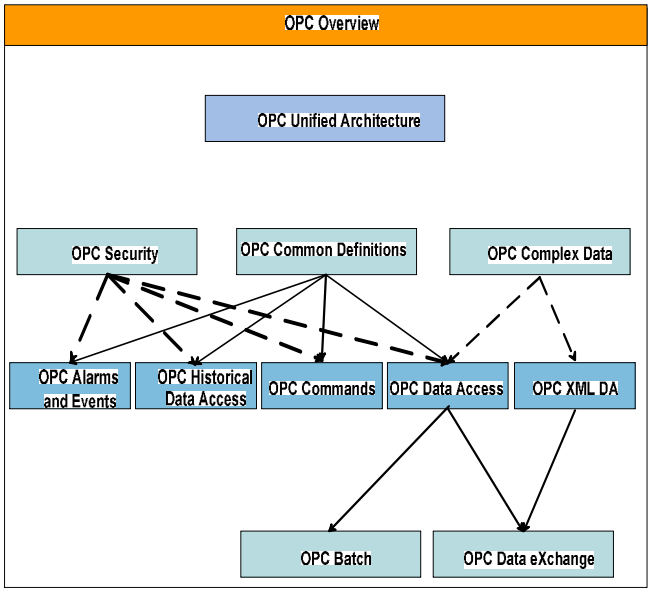


**Figure 3.** OPC Overview [18].

### OPC Data Access

**OPC Data Access (OPC DA)** was the first solution developed by OPC Foundation, within Middleware. In 1996, the first version was commercialized, meaning OPC Data Access 1.0. This offered well defined specifications conditions, including Device model, Data model and Communications model. It also used DCOM middleware, and the basic level accepted is related to the user.

An OPC DA (Server) allows to OPC DA (Clients) customers to receive information about different objects: server, group or elements.

When an OPC server is available for a device, that device can be easily used with a SCADA system. This system uses a single protocol, so as to access a series of devices.

The OPC object stores information about server and acts as a recipient to the OPC group. On its turn, the OPC group stores data about itself and foresees a mechanism of organizing the OPC elements. The OPC elements are represented by the connections to sources of data used by the server.

Within OPC DA specification conditions, two interfaces of reading/writing, meaning one for synchronous module and one for the asynchronous module.

Concerning the synchronous module, the client can read synchronous from cache. This interface is compliant to simple customers, which use few data. As conclusion, it is simple and reasonable, taken into account the efficiency point of view.

As regards the asynchronous module, the client can overwrite the data stored in cache, by using IAdviseSink or IOPCDataCallback. This aspect seems to be more complex, but more efficient. The asynchronous access is recommended, since it minimizes the use of CPU and resources.

To all situations, the OPC DA server offers access to client, for all current values of OPC elements. It owns only current information in cache. The old information is overwritten. As result, the case where an OPC DA client will take over all modifications of values (in asynchronous module) is not ensured. For such cases, two more OPC specifications are also defined. These are represented by OPC Alarms & Events and OPC Historical Data Access.[4]

### OPC Alarms & Events

**OPC Alarms & Events (OPC AE)** interface is provided with a mechanism, and when new events occur or the alarm state is switched on, the mechanism will inform OPC AE client. In the same time, OPC AE clients are allowed to set up some lists of events, the conditions which will be supported by OPC AE server, as well as the achievement of their current status.

An alarm signifies an abnormal condition, which occurs in special cases. Such special cases assume conditions. In the same time, this condition is a state of OPC Event Server or a state related to objects included within. As states of OPC Event Server, one might emphasize: HighAlarm, HighHighAlarm, Normal, LowAlarm and LowLowAlarm.

Instead, an event means an occurrence, which can be detectable by OPC AE server or OPC AE clients. In the

same time, an event can be associated or not, but depending on conditions. Though, there are some events not connected by specific conditions, such as: actions of the operator, modifications of system's configuration and system's errors.

The specification conditions of OPC AE offer some methods by which an OPC AE client is able to determine different types of events accepted by OPC AE server, to introduce specific events, so that OPC AE clients can receive notifications of their events and to access the conditions implemented by OPC AE server. The filters can be used so as to define a subset of events taken into account.[3]

### OPC Historical Data Access (HDA)

Most of historic systems have produced sources of information, which should be distributed by users of software. They are provided with own interfaces, so as to disseminate data. The historic systems are divided in: simple Trend data servers, complex servers of data compression and analysis. [5]

### OPC XML-Data Access

Launched in 2003, **OPC XML-Data Access (OPC XML-DA)** represents the adoption of XML technologies set by the OPC Foundation, so as to facilitate the data exchange on the Internet.

OPC XML DA specification conditions are similar to that of OPC DA, except the communication model.

OPC XML-DA is based upon interfaces that simplify the data exchange between different levels of an hierarchy and for a wide area of platforms. The aim of these specifications is of helping support for: OPC Data Access 2.0x/3.0 data HTTP and SOAP, services based upon overwriting and approaching the security.

In order to establish the performances and steadiness, more tests were accomplished, starting from writing the OPC XML and using a set of tools (toolkit); in the same time, the necessary time so as to read OPC elements is taken into account. These tests were accomplished on both platforms (Windows and Linux). The result of these tests proved that only about 6,000 elements per second are read when using OPC XML-DA, and about 76,000 elements are read when using OPC on Linux. The main problem was that an interface's definition is not always leading to a correct code.[7][8][9][10]

### OPC Unified Architecture

**OPC Unified Architecture (OPC UA)** was introduced in 2005, but this wasn't launched in the same year. OPC UA represents a conjugation of OPC interfaces. OPC UA carries out secure, reliable and efficient services.

OPC UA is an independent platform standard, by which various types of systems and devices can communicate by means of messages between clients and servers, no matter the networks types. In the same time, OPC UA was implemented so as to resist attacks. The secure and robust communication ensures the identity of clients and servers. The OPC UA servers defined object models, on which clients discover dynamically.

The servers can access both current and historic data, as well as OPC Alarms & Events, thus announcing the clients about the modifications came out.

OPC UA can be mapped by a variety of communication protocols, and the data can be encoded on different modes, in order to not compromise the portability and efficiency. [7][8][9][10][11][12][16]

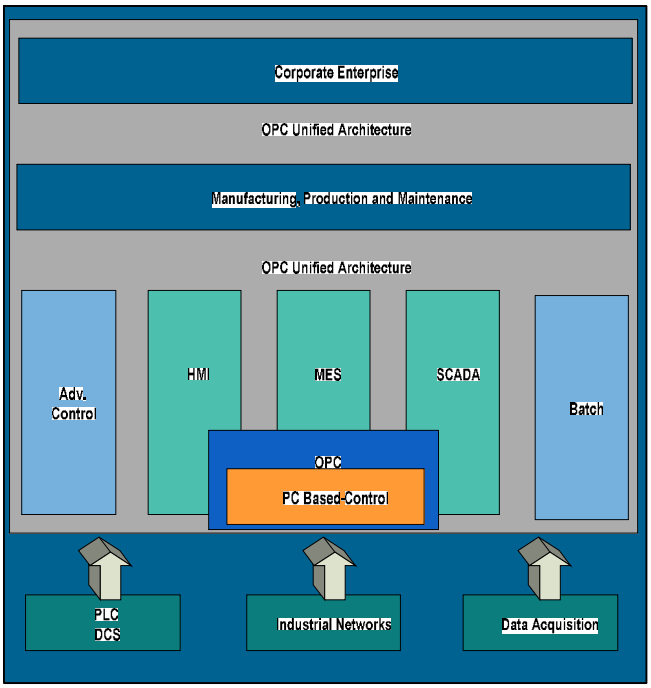


**Figure 4.** OPC Unified Architecture [18].

### OPC Express Interface (OPC XI)

**OPC Express Interface (OPC XI)** represents a technology added to OPC Foundation portfolio, in completing the OPC Unified Architecture (OPC UA) technologies and of the classical OPC based upon COM.

The OPC XI interface is the result of some companies' collaboration within process industry, which provides OPC, in the wish of developing a secure solution and easy to integrate to a wide variety of communications.

The main OPC XI objective consisted in creating a migration, from the classical OPC to that based on .NET. Moreover, OPC Xi can be used as NET WCF standard interface, so as to develop the new OPC servers.

The main characteristics of OPC XI are: Security – secured communications, including for the access by means of firewalls; Simplicity – determines that servers and clients to be easy to implement and configured; Robustness – trust communications, with recovering of errors; Plug-and-Play – automatic discovering of servers and their communication capacities; Reverse compatibility – accessing of OPC DA, AE and of HAD servers, by using .NET as single interface; Transparency of Protocol – the client/server model is efficient when both the client and server are on the same platform or within the same LAN network, when the client is situated within extern networks. [13][14][15][16][17]

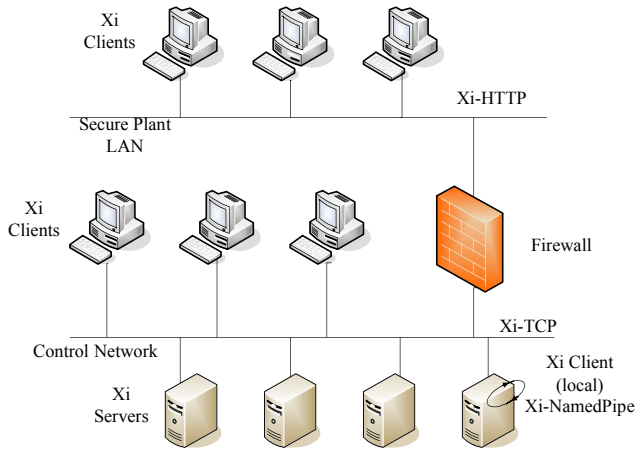


**Figure 5.** OPC Express Interface (Xi) [17].

## IV. MIGRATION STRATEGIES TO OPC UA APPLICATIONS

An essential topic refers to the way of migration, from the old OPC server, based upon COM, and the new Web-service, based upon the new technology, "Unified Architecture". The OPC servers based on COM used a property interface, so as to accessing data about the devices of the control network.

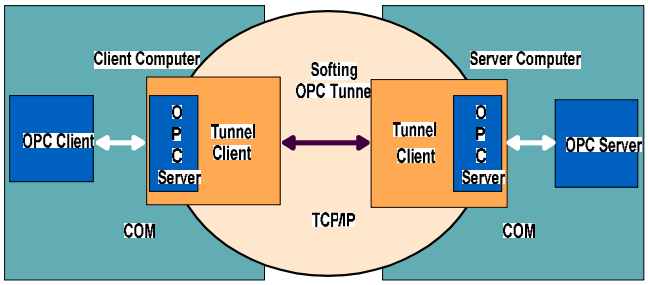


**Figure 6.** OPC and DCOM [20].

There are two approaches that can take into consideration, meaning: packaging the already existing servers and direct accessing of the data device; these approaches are illustrated in Fig. 7.

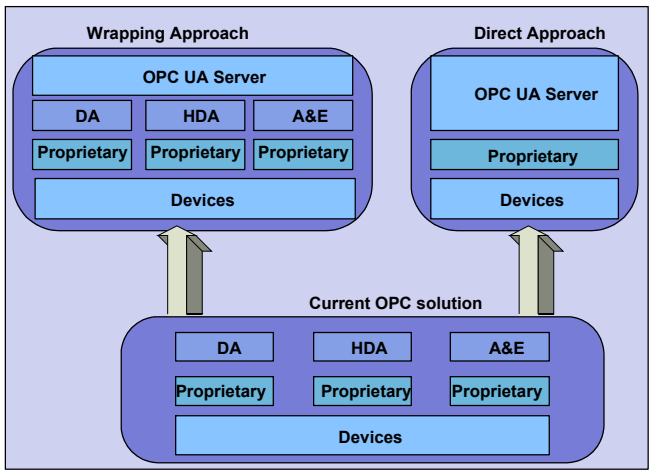


**Figure 7.** Migration strategies [2].

Packaging the existing servers might reduce the developing time of implementing OPC UA servers, since the existing OPC servers can be used. [17][18][19][20]

## V. CONCLUSION

OPC UA signifies an important phase, so as to integrate the new technologies and concepts within industrial applications. Standardization of services oriented architecture and unifying various types of OPC servers will open new application areas.

The OPC specifications contribute to simplifying the integration of other applications, such as HMI or SCADA systems.

More other standards get the better of OPC UA, so as to define specifications for their specific areas, information available by means of OPC UA.

OPC XML-DA and OPC UA are no longer based upon DCOM, but on services oriented architectures (SOA).

The OPC XI interface is the result of some companies' collaboration within process industry, which provides OPC, in the wish of developing a secure solution and easy to integrate to a wide variety of communications.

We propose a new project development in the near future, so as to bring improvements to the OPC server.

## REFERENCES

[1] Brown, N., Kindel, C.: "Distributed Component Object Model Protocol - DCOM/1.0". Microsoft Corp., 1998.
[2] Stefan-Helmut Leitner, Wolfgang Mahnke: "OPC UA – Service-oriented Architecture for Industrial Applications", Germany.
[3] OPC Foundation: "Alarms and Events Custom Interface Standard". Version 1.10, Oct. 02, 2002.
[4] OPC Foundation: "Data Access Custom Interface Standard". Version 3.00, March 04, 2003.
[5] OPC Foundation: "OPC Historical Data Access Specification". Version 1.20, Dec. 10, 2003.
[6] Microsoft Corporation: "Windows Communication Foundation", September 2006.
[7] OPC Foundation: "OPC UA Specification: Part 1 – Concepts". Version 1.00, July 28, 2006.
[8] OPC Foundation: "OPC UA Specification: Part 2 – Security Model". Version 1.00, July 28, 2006.
[9] OPC Foundation: "OPC UA Specification: Part 3 – Address Space Model". Version 1.00, July 28, 2006.
[10] OPC Foundation: "OPC UA Specification: Part 4 – Services". Version 1.00, July 28, 2006.
[11] OPC Foundation: "OPC UA Specification: Part 5 – Information Model". Version 1.00, July 28, 2006.
[12] OPC Foundation: "OPC UA Specification: Part 6 – Mapping". Release Candidate 0.93, June 01, 2006.
[13] OPC Foundation: "OPC UA Specification: Part 7 – Profiles". Draft 0.93, July 28, 2006.
[14] OPC Foundation: "OPC UA Specification: Part 8 – Data Access". Version 1.00, Sep. 25, 2006.
[15] OPC Foundation: "OPC UA Specification: Part 9 – Alarms and Conditions". Draft 0.75, June 28, 2006.
[16] OPC Foundation: "OPC UA Specification: Part 10 – Programs". Draft 0.4, Sep. 01, 2006.
[17] www.opcfoundation.com.
[18] www.softing.com.
[19] www.moxa.com.
[20] www.automation.com.