OPC for Process Maintenance

SCHWARZ M.H.[1], BOERCSOEK J.[1,2]
[1]University of Kassel
Department of Computer-architecture and System-programming
Wilhelmshoeher Allee, 34121 Kassel
Germany

[2]HIMA Paul Hildebrandt GmbH + Co KG
Albert-Bassermann-Str. 28, 68782 Brühl
GERMANY

Abstract: - This paper reports on the authors' experience in process maintenance and networks and presents recent research and development approaches within the department of computer architecture and system programming at the University of Kassel. The presented approaches are based on well established OPC client-server architectures. The OPC standard is accepted in industries and academia.

Key-words: OLE for process control, OPC, client-server architecture, condition based maintenance,

1 Introduction on OPC

Over the past several decades, few trends and developments were significant: closed-loop control, micro-controllers, Ethernet based communication and the latest, is the movement towards predictive and condition based maintenance and monitoring [8, 14, 15].
However, it is still astonishing and surprising that many control loops are being operated in manual mode, or are detuned. Maintenance is carried out at fixed time intervals not when it is necessary (e.g. not depending on the working hours of the equipment) and process data is only stored and often not used to optimise processes. [14]
Furthermore, the actual computer system such as hardware, network, switches and so on, are normally not monitored, it is assumed that they are always operating correctly [15]. New and modern strategies do incorporate these components as well, then the best control and monitoring system is worthless if the information cannot reach the operator due to a malfunctioning switch [14].
This papers concentrates on flow and pressure control loops based on the ubiquitous control structure, compose more than 90% of the loops traditionally encountered in the manufacture of chemicals and pharmaceutical products [12]. Techniques to improve the efficiency of the tuning process are always of interest. This paper also focuses on the usage of OPC server and client architecture to establish the necessary network.
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 [18] 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 areas. 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 [6, 10] 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 [18].

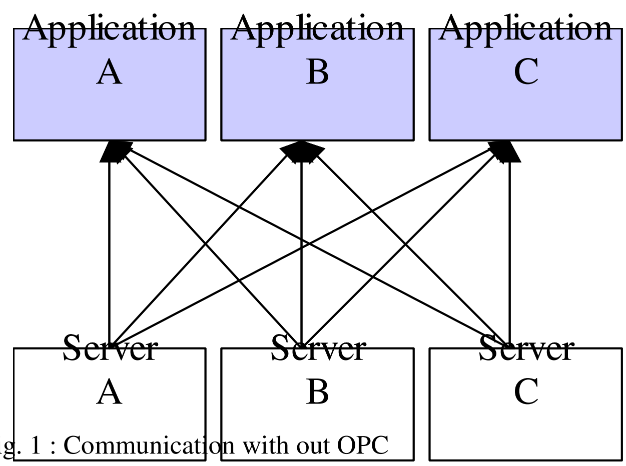


Fig. 1 : Communication with out OPC

If the manufacturers made changes on the bus systems, bus nodes or protocols, then system drivers had to be adjusted and re-tested, this is cost intensive and provides a high possibility of errors [4, 10]. 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 [6, 10]. 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 [10] 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) [6,17]. 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.

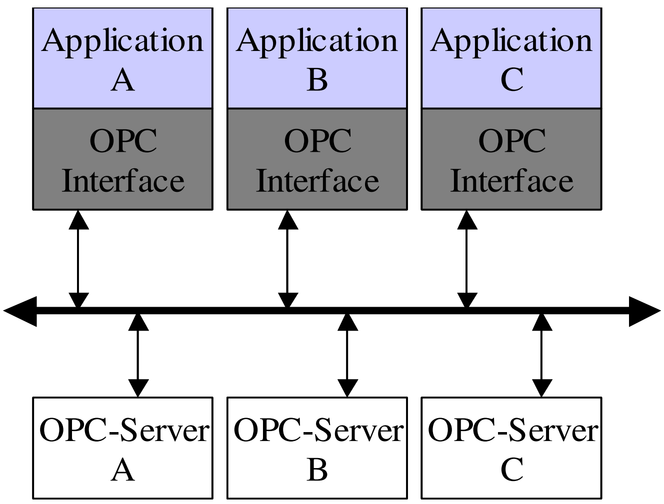


Fig. 2: Communication with OPC

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.
The remaining paper is structured as follows: Section 2 summaries briefly the different OPC standards and describes for which applications they can be used. Section 3 presents strategies for the development to incorporate IT maintenance into the system. Section 4 develops strategies for the implementation of condition base maintenance using OPC. Section 5 summaries the strategies and gives an overview of future work of developments in the area of condition based maintenance within the department.

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 [6, 10,16]. 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 [10, 16].

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 an event has occurred. It is important to note that AE server and DA server can have the same data sources [10, 16]. 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 [10]. 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 [6, 10, 16, 17].

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 [10,14]:

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 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 [10]. 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 is within the range of 100-1000 variables. The number of process variables supplied by a HDA-server is within 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 [10, 16, 17].

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 [10]. 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 [6, 10, 16].

3 OPC for the IT Maintenance

Most modern control rooms are connected with the operating PLCs (Programmable Logic Controller) via reliable Ethernet based networks including cabling, switches and routers [4,11,13]. Additionally, many engineering systems, control systems and applications are operating on standard computer hardware, mainly based on Microsoft's operating system. Those companies also have process relevant equipment such as pumps, compressors, sensor, actuators or boilers and compressors [15]. It is the maintenance or mechanical department to make the move towards condition based maintenance. The idea is simple: collect reliable, good and accurate data from the system and each component and use this information to allocate problems or unusual behaviour and maintain, repair or exchange the component before it gets to dangerous failures [4,5,15]. Therefore, chemical, pharmaceutical and process industries are also starting to look into the approach of collecting and evaluate vibrating, rotating, thermal and other mechanical assets [15]. However, in most cases, the information technology (IT) is not considered for maintenance, which includes the hardware and software for collecting, transmitting, storing and analysing data. The reasons are manifold, standard hardware is not considered as expensive and often non-critical, a lack of communication between engineers onsite and computer engineers, but mostly nobody is aware of the fact the IT equipment should be considered in the maintenance procedures [5,15].

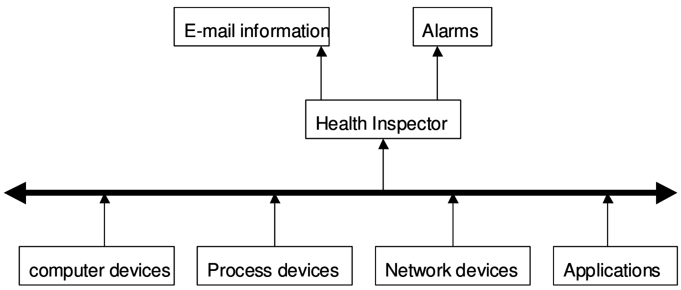


Fig. 3: Health inspector

Every device is sending its health state or performance status to the health inspector, which can be a normal OPC server-client configuration and the health inspector decides if further actions are necessary, for instance an alarm is set up or the maintenance or IT department receives a notification that a certain device needs to be maintained. The OPC server collects the necessary data from using a Simple Network Management Protocol (SNMP) to communicate with each device or with performance manager from of each PC [15].

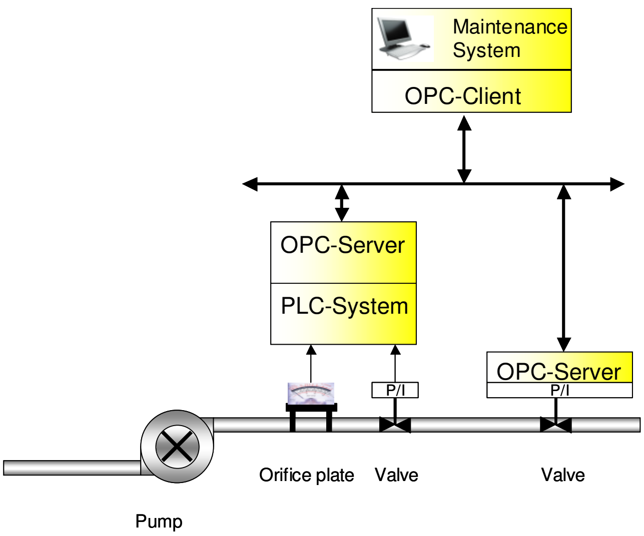
4 Process Condition Maintenance and Monitoring with OPC

4.1 Automatic data acquisition

One of the key issues in chemical, pharmaceutical and process industries over the last few years is the movement towards predictive and condition based maintenance. The idea is simple: use the real-time information from the sources and equipment to find out how healthy the system is, and maintain it when it is needed and the maintenance department is informed [8]. Normally, this follows four steps:

1. The data is acquired from a SCADA system
2. The data is analysed and stored
3. A decision system uses the data to determine the health state of the system
4. A decision system informs the maintenance department and schedules the work.

Software applications are using OPC communication to collect real-time data from the devices and systems and provide this information to external systems. Automating the data collection, results also in a decrease of amount of faulty data and information. Until now, manual data collection is still practice in many companies, which results in unreliable data [8]. Then the normal procedure would be that a worker reads the data from meters or measure the health of the equipment with a device and records the data [8]. Afterwards, the data is put into a data base or maintenance system. From this example it can be seen that this procedure is fault-prone [8] .
Therefore, the movement is towards fully electronic data collection and recording, which has also the advantage that all data are in the original state from each automation system and all data points possess an exact time stamp.



. Fig. 4: Collecting data for maintenance system

Figure 4 shows a system for data collection to be used within a maintenance system. The data is sent to the maintenance system either via a PLC, which possesses a OPC server, or new and modern devices are already equipped with an OPC server to communicate with the client of the maintenance system [8]. The advantage of new and modern sensors will be discussed within a separate section.

4.2 Condition based maintenance

After the communication and the capture or real time data for a maintenance system is set up, then the effort should be concentrated to develop predictive and condition based maintenance methods.
Traditionally, maintenance is scheduled on fixed periodic intervals and it does not matter if the equipment was used or the state of actual conditions. This means equipment that is rarely used is going to be maintained too often and frequently used systems might be maintained not often enough [8]. Also, the condition under which the equipment is operating is not taken into account. A system which has to operate in a harsh environment might be maintained as little as a system which operates under good or ideal conditions. The likelihood rises that systems which are suffering from not being maintained are going to fail and can result in expensive repair, exchange of components or a loss of production. This could have been avoided if the system had been monitored appropriately and even more important necessary maintenances are ignored which should have been carried out immediately.
Industries realises that it is important to measure und analyse important process data and to indicate the process performance and its conditions.

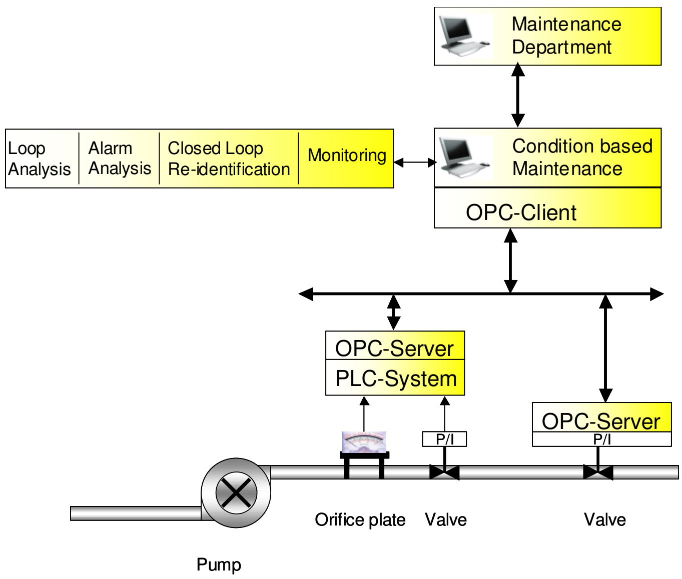


Fig. 5: Modern Maintenance System

On important variable that can indicate the condition of the equipment is the runtime hours. It is not difficult to acquire this data, by setting a counter to calculate the working hours. When the counter oversteps a predefined value a condition based maintenance programme alerts the operator or informs automatically the mechanical or maintenance department [8].
Figure 5 shows a modern maintenance system. Real-time data is collected and the condition based maintenance system uses the data to make decision such as analysing the loop, sending out an alarm, re-identify a system or monitor it. If necessary, the maintenance department is informed to carry out the maintenance of components or equipment [8].
Additional information and states are necessary and should be taken into account when decisions have to be made. Important variables are calculated by carrying out vibration analysis, measuring the temperature, flow rate measurements, consistency of lubricants. These variables should be considered, and therefore maintenance might be carried out, much earlier then the runtime hours would suggest. Therefore, the environment in which the system is operating is also considered. All these factors and variables are important when the health of a component, equipment or system has to be estimated.

4.3 Identifying malfunctioning instruments

If an instrument gets broken, then it is simple to identify it and repair or exchange it. But often, it is a process when an instrument start to get malfunctioning, maybe it works in certain ranges correctly but in others not. So, it is very difficult for the operators to recognise it that the instrument is not operating correctly. But, when alarms are set and counted how often and how long the alarm occurred then an operator is getting informed when the predefined limits are reached and workers can check, recalibrate, repair or exchange the instrument even before the operations are aware of the problem. This eliminates potential hazard, which might have resulted in unscheduled shutdowns.

4.4 Monitoring multiple factors

The advantage of modern maintenance system is that several variables and factors can be captured, estimated and calculated to determine the health state of a system. But also advanced sensors are necessary or can be useful in order to estimate the current state.
Oil degradation is the true "root cause" of many serious machine failures caused by wear and internal oil contamination. In addition, the optimisation of the service life of oil is a secondary but important consideration from economic and environmental perspectives [1]. Periodic off-line laboratory oil analysis may not provide sufficient early warning of degradation and contamination. There is a requirement for new generation sensors for real-time oil analysis for comprehensive characterisation of oil lubrication effectiveness.
The on-line micro-sensor monitors the quality of lubricating oil in compressors. The main features of the sensor are an optical grating, micro-optical components and a detection system assembled to two spectrometer systems, one for the visible and one for the infrared spectral region [1].
The primary objective is the development of a micro-sensor system to monitor, in real-time, the quality of lubricating oil. The typical target are compressors, however, this is envisaged to be extended to other machine types. This requires the integration of micro-technologies and new methodologies to improve control and monitoring systems on mechanical devices.
The sensor systems utilises both visible and infrared spectral ranges. The information contained within these spectral bands is huge and hence an efficient means of condensing the data into its most useful parts is an essential part of the overall system. This is where artificial intelligence and intelligent software systems are required [1].
Data has been collected from a programme of oil sampling from end user systems, from the analysis of historical data within end-user data bases, and also from controlled laboratory trials. The spectral information measured and the oil quality analysis results have been used to develop degradation detection models. These models have been recursively pruned to allow them to be used with data from the micro-sensor without significant loss of accuracy [1].
While it was necessary in the past to build the control-units onto the sensor, with a OPC- client-server architecture the sensors is only measuring the values and the analyses is done by the maintenance system. Therefore, complex software has not be squeezed onto a small micro-controller system

4.5 Maintaining control loops

A recent approach of a modern maintenance system is to consider and inspect the performance of control and regulation loops. Recent surveys [8,12] showed that about 80% of the control loops in industries need maintenance or a re-identification or re-tuning. 20% of these control loops are in a critical state and have a significant impact on the process performance. Unfortunately, most operators or maintenance personnel decide either to further detune the controller or switch them to manual mode, which is contradictive to the idea of control and regulation.
By continuously monitoring the performance of the control-loop, especially those which are critical, the work load and pressure on the maintenance department is reduced.

A further approach is to develop and implement automatic identification or tuning methods [12]. Within in the last few years, process identification move towards closed-loop identification [12, 19]. This would mean that maintenance department is not involved, when the maintenance system or observer detects that a control loop is not performing as it should be. A closed-loop identification is initiated and re-identifies the system and calculates the new parameters for the controllers. The advantage is that this can be done while the system is operating, and the plant has not be shutdown as traditional open-loop methods require. For retuning a simple but very effective approach is the relay + integrator method suggested by Åström and Hägglund in 1994 and 1995 [2,3].
In 1995 they demonstrated that it is only necessary to replace the gain of controller by a relay to determine the critical parameters K_u (the ultimate gain) and the period of the oscillation, defined as P_u (the ultimate period). After a transient phase, this (normally) causes the system to oscillate with a fixed frequency, constant amplitude 'sinusoid' known as a limit cycle. When using describing function theory [7], the control parameters can be estimated as demonstrated [2, 3, 12]. They basically turned the Ziegler-Nichols open-loop method into a closed-loop technique. The more famous method replaced a controller by a relay + integrator [3]. This introduced a design choice since increasing phase margin is usually accompanied by improved relative stability.

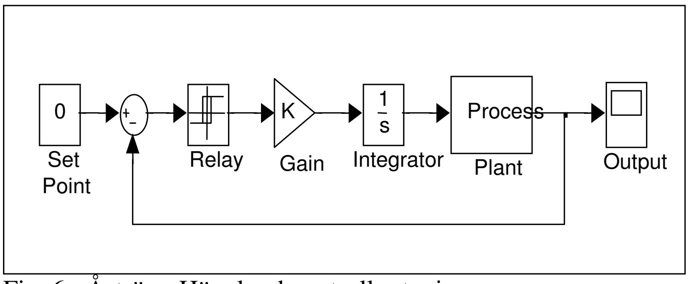


Fig. 6: Åström–Hägglund controller tuning

Other identification methods to model a process with relay methods can be found in "Relay Feedback Analysis, Identification and Control" by Wang [19], where the transient phase is exploited to estimate the model. In general, it has to be decided, if the control structure onsite are equipped with the necessary identification and tuning methods, or these methods are implemented on the maintenance system. In the latter case the communication networks is part of the model.

5 Summary

This paper described the current development of maintenance systems and the authors' experiences in process maintenance and networks. OPC presents a chance for many University departments to get their research results onto the market. This paper demonstrated that mature and well established research areas can be combined with new techniques and can result in powerful tools for process industries.
The current development incorporates different identification and tuning methods into the system and to establish a reliable and maintenance system. After the main integrations will be finished, the system has to be analysed to develop and establish a safe and reliable system [4, 5] according to the international standards such as the IEC 61508 [9].

References:

[1] Adgar A., Schwarz M.H. & MacIntyre J. (2004) Development of Intelligent Software for a Micro-Sensor based Oil Quality Analysis System, *COMADOM 2004*, Cambridge, United Kingdom
[2] Åström K., and Hägglund T. (1995) 'PID Controllers: Theory, Design and Tuning.' Instrument Society of America 67 Alexander Drive,PO Box 12277, Research Triangle Park, North Carolina 27709,USA
[3] Åström, K. & Hägglund, T. (1984), 'Automatic tuning of simple regulators with specifications on phase and amplitude margins.' Automatica, Vol 20, pp 645-651.
[4] Börcsök J., *Elektronic Safety Systems*, Hüthig, 2004.
[5] Börcsök J., *Functional Safety*, Hüthig, 2007
[6] Chisholm, A. DCOM, OPC and Performance Issues, *Intellution Inc*. 1998
[7] Gelb A. & Vander Velde, W.E. (1968) 'Multiple-input describing functions and nonlinear system design'. McGraw-Hill.
[8] Gould J. CMMS: Integrating Real-Time Information for Condition-Based Maintenance. *Matrikon, Inc.* 2005
[9] IEC 61508, International Standard 61508: *Functional safety of electri-cal/electronic/programmable electronic safety-related systems*, Geneva, International Electrotechnical Commission, 2000
[10] Iwanitz F., Lange J. OPC Fundamentals, Implementation and Application, 2nd Rev.-Edition *Oldenburg*.2002
[11] Kondor R., OPC, XML, .NET and Real-Time Applications, *Matrikon, Inc*. 2007
[12] Leva A., Cox C., Ruano A. IFAC Professional Brief: Hands-on PID autotuning: a guide to better utilisation, International Federation of Automatic Control
[13] Michaud A. Creating Secure OPC Architectures, *Matrikon, Inc*. 2007

[14] Mitchell W. Historical OPC HAD: Helping Improve Plant Performance. *Matrikon, Inc*. 2003
[15] Murphy E. OPC Maintaining Healthy IT Assets. *Matrikon Inc*. 2006
[16] OPC –Foundation OPC Overview OPC –foundation 1998
[17] OPC –Foundation OPC United Architecture, *OPC –foundation* 2006
[18] Reissenweber, B. Feldbussysteme, *Oldenburg*.1998
[19] Wang, Quing-Guo, Lee, Tong H., Chong, Lin, *Relay Feedback Analysis, Identification and Control*, Springer 2003