Advances of OPC Client Server Architectures for Maintenance Strategies – a Research and Development Area not only for Industries

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
{m.schwarz, j.boercsoek}@uni-kassel.de

Abstract: - OPC is a matured and accepted standard in industries, but hardly used in industrial applications developed at universities, especially in Europe. Publications in this area, either using OPC-communication or developing applications have been hardly increased over the past several years. This paper reports the current state of the standard, but also describes current problems and how they can be managed and overcome. The authors also give an overview of the recently published new standard, which is called: OPC unified approach. After the fundamentals are described in this paper, the authors present their experiences with OPC to combine it with methods and procedures to develop a modern maintenance system. It also reports current developments at the Department of Computer Architecture and System Programming at the University of Kassel.

Key-Words: OLE for process control, OPC, client-server architecture, condition based maintenance, unified architecture, OPC unified approach, OPC-UA

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

which is the latest movement and trend [12, 25, 28, 29]. 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 [25]. Furthermore, the actual computer system such as hardware, network, and switches etc., are normally not monitored. It is assumed that they are always operating correctly [28]. New and modern strategies do incorporate these hardware components in the maintenance strategy as well, then the best control and monitoring system is worthless if the information cannot reach the operator due to a malfunctioning switch or component [25].

This paper concentrates on flow and pressure control loops based on ubiquitous control structures, which compose more than 90% of the loops traditionally encountered in the manufacture of chemicals and pharmaceutical products [20]. 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 communication.

However, 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 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 specific 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 in journals is not increasing, especially in Europe. Asian countries have already understood the benefits of OPC. 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 onto 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. Within few years, over [33] 50 different network systems were designed for different applications. Few vanished into thin air, but most are only used in specific application areas. And few, mostly developed by big companies, have become standard, which are not always the best ones.

However, with the establishment of a number of different networks for different applications 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, but process industries have started to mix them in the processes. Additionally, for every single bus system, manufacturers have to develop drivers and maintain [8, 15] 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 [33].
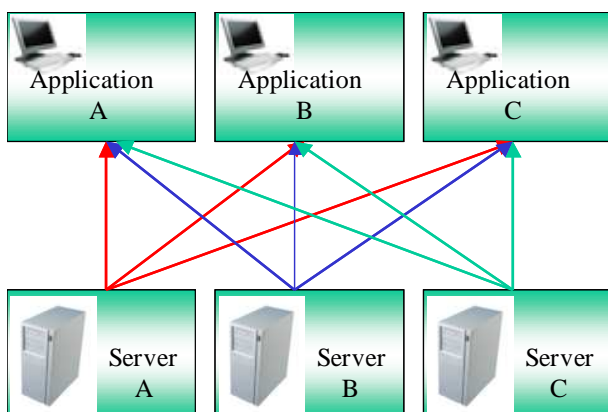

Fig. 1 : Communication with out OPC

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 [4,15]. 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 [8,15]. Universities and institutes had basically no possibility 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 immense 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 [15] 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 system Windows. The standard was named OLE for Process Control (OPC) [8,30]. It was ensured that an open participation was possible by incorporating feedback from various companies all over the world and obtaining acceptance from both industries and end-users. 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.
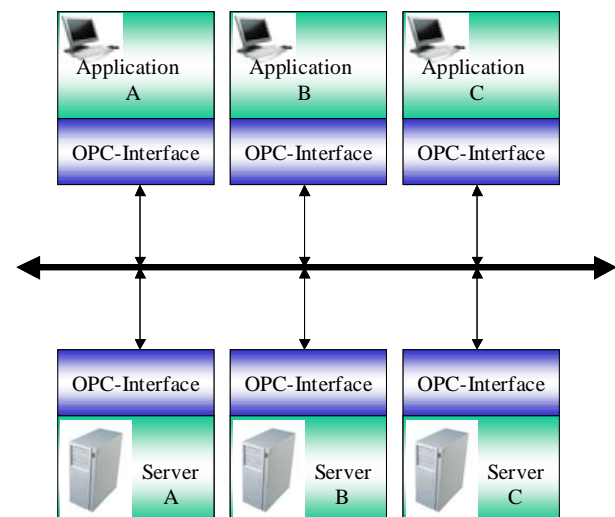

Fig. 2: Communication with OPC

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 current problems with OPC-standards, such as using COM/DCOM

methods and discusses security issues. It also describes the new standard OPC-UA and XML-web-services. Section 4 details different strategies in order to implement OPC in a redundant strategy to improve reliability and availability. Section 5 shows strategies for the development to incorporate IT maintenance into the system. Section 6 develops strategies for the implementation of condition base maintenance using OPC. Section 7 gives an overview of the planned developments within the department using OPC. Finally, conclusions are drawn in section 8.

2 OPC

The OPC foundation released several specifications for different data communications on the basis 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 [8,15,30]. 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 be 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 [15, 30].

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 [15, 30]. The difference is that a DA-server provides a

continuous data stream, the AE-server on states changes. The automated transition of values can be accommodated by a relative change of the value. This adjustment is only possible for analog values [15]. An AE server does not sent process values to a client, but the information that something happened or occurred, e.g. a valve has been opened, or a temperature of a process 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 an e.g. DA-server [8, 15, 30, 31] and has to made its decision by its own.

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 unit can have the privileges of readable, writable and changeable. Two different HDA-server client implementations exist [15,25]:
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 evaluate it, for instance it computes the mean value, and minimum or maximum 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 [15]. A HDA server is similar to a DA server, but a HDA server does not have any objects such as OPC-groups or OPC-Items. 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 [15,30,31].

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. Subscription data is sent and report information is received. Products for batch processing have to be manufactured according to the IEC 61512-1 [15]. 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 [8, 15, 30].

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 [17]. 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 [8,17], especially when using the new XML web services. Many managers and process engineers fear that the OPC-DA and OPC-AE server and clients will be soon outdated and think about the point in time when they should swap technologies [8,17,32]. The figure below shows the different changes, which leads to the new OPC standard.
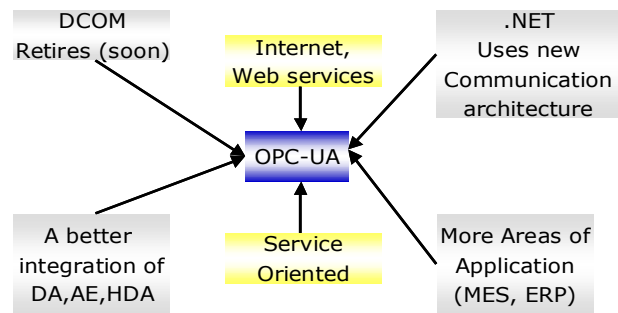

Fig. 3: COM / DCOM

3.1 COM / DCOM Technology

OPC communication is based on Microsoft's COM/DCOM technology. When OPC applications are installed on one PC, then they are using Microsoft's COM technology (Component Object Model) to exchange data. But when OPC applications are installed on two separate PCs, then they are using Microsoft's DCOM (Distributed Component Object Model). COM messages are then basically wrapped in a Microsoft security layer, called DCOM [13,17,18]. 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 and special action and preparations have to be taken [17]. 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 [17]. 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. For users, it is not possible to change the six minutes time [17]. The demanded application just waits for DCOM to answer. All process data is unavailable during this time. And just imagine this application is a controller device which needs several data points per second to calculate the appropriate controller output to be sent to the process. Software developers have to build a monitoring device around the DCOM communication to observe that it is functioning correctly [17].

XML web services are 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 [17].

Another important issue is that COM/DCOM is based on Microsoft operating system. Other operating systems 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 [13,15,17].

3.2 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 [17,21,23]. XML web services are based on XML and is 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 frameworks. 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 4 shows the general concept of XML web services [17,23].
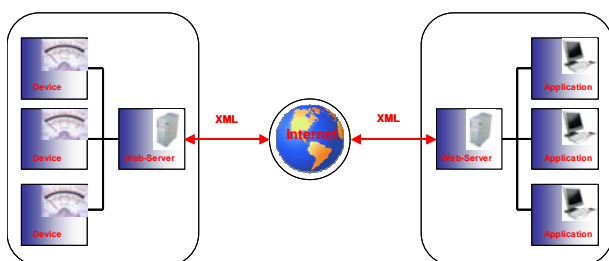

Fig. 4: Concept of XML web services

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 for the time interval where 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 applications 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 suitable for process real time applications [17]. However, XML-web services are a very new technology for applications in process and automation applications, therefore, these problems will be solved in the near future and the present limitations will vanish.

3.3 OPC Unified Architecture (UA)

OPC Unified Architecture [15,17,31] will currently not 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-HDA [30,31]. 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 into account security aspects. 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 defines 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 intends 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 [30,31].

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 of a server or client. Since a whole description of this standard would be far beyond of the scope of

this paper, the authors refer to the new standard [15].

3.4 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 [18,24,26]. 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 [18,24,27]. 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 reasons why the OPC foundation selected DCOM as the base for OPC communication [18,24,27]. DCOM requests many ports for locating other hosts, resolving names, sending data. If these ports are unavailable, then DCOM automatically starts to search for others. All services and ports used by DCOM are targets for hackers as well. If the virus has 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 [18,24,27]. 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 data and controls the privileges and authorisations for clients using DCOM. This would separate the server from direct access by any client [18,24,27]. Another and probably most promising way is to tunnel the data from the OPC-server to the OPC-client. Companies such as MatrikonOPC [18] 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 removed from the systems [18,24,27]. 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.
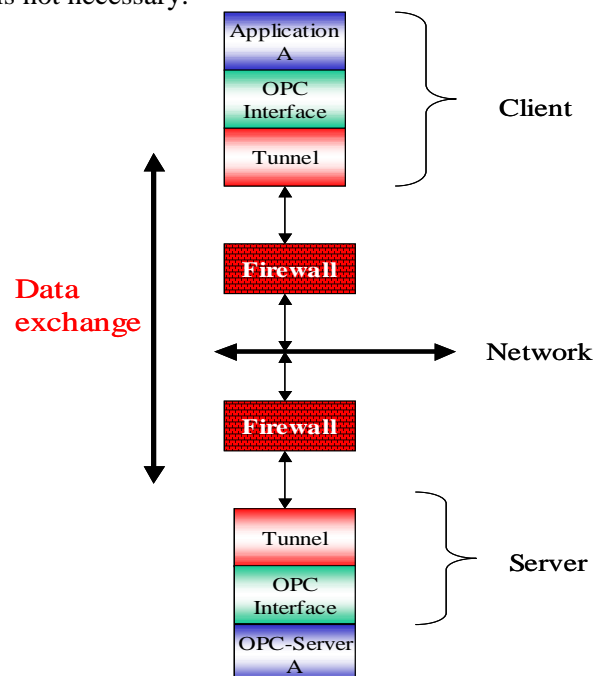

Fig. 5: Secure communication

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

4 Redundant System Approaches

In many industrial applications for process and automation industries, redundancy is an important feature to increase the efficiency and reliability of the system [24,26]. 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 breaks down while link based failures occur when the physical link to the devices, control units or systems breaks. From this view point three different redundancy strategies exist and are listed below. Figure 6 shows all different redundant strategies.

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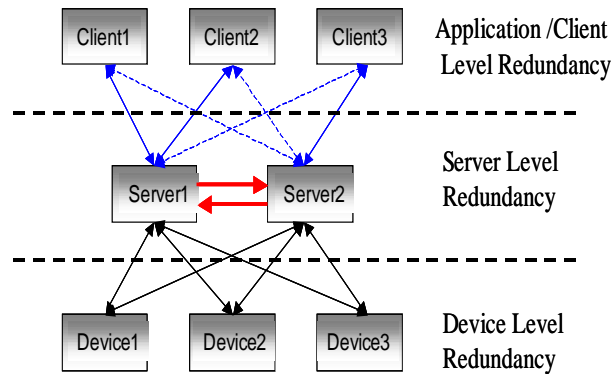- Application / client Level Redundancy

Fig. 6: Different redundant strategies

4.1 Device Level Redundancy
In a Device Level Redundancy strategy, controllers or data collection devices are implemented in a redundant configuration. If the server-device connection fails, then the redundant device starts to operate. In this strategy no client is involved. Vendors start to provide OPC server with redundant device support or redundant communication channel support [26,32].

4.2 Server Level Redundancy
Server Level Redundancy is available when two servers provide data to a client. One can be the primary server the other one 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 the first server and client fails, then 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 7 [26,32].
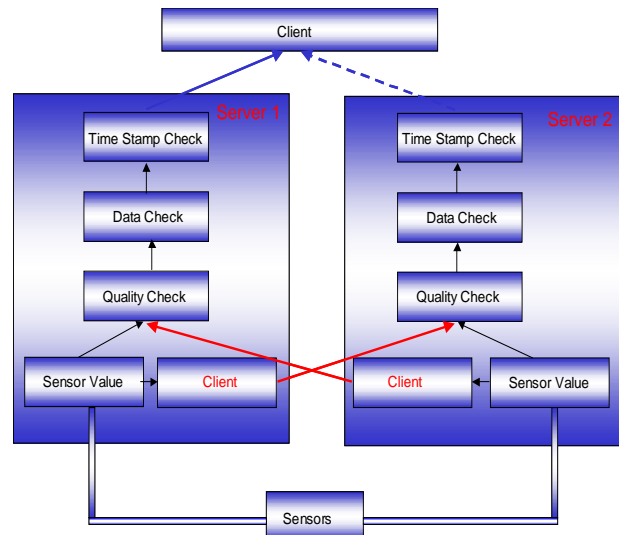

Fig. 7: Redundant OPC server structure with an additional client to exchange process values

4.3 Application or Client Level Redundancy
Application or Client Level Redundancy exists, when two clients or applications are implemented in a redundant way. If the connection or link to a server or the application itself fails then the second client starts operating. Figure 8 shows the principle of a redundant client application. The client receives from two servers process data, its quality and time stamp. If one of the free values differs, then the client informs an operator, or sends out an alarm. In many applications it might not be necessary to have a link to two servers during normal operations, then one server provides the process information the second one is in standby mode [32]. From Figure 8, it can also be seen that a normal client is not satisfactory for being used in a redundant approach, as it compares the data from two servers, which a normal client would not do.
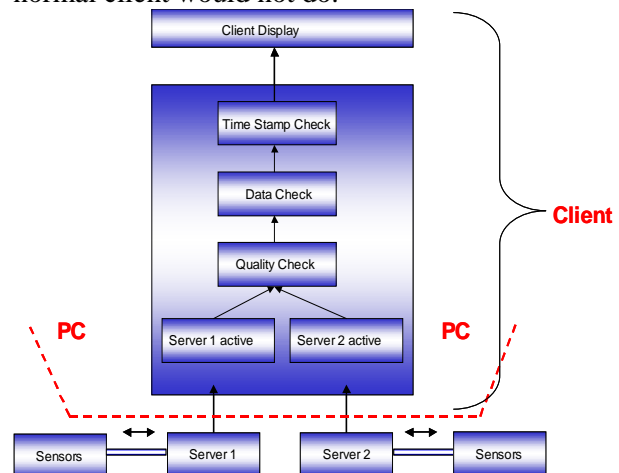

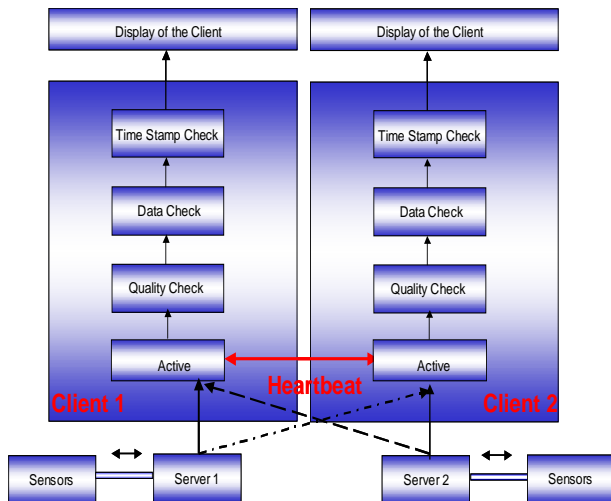Fig. 8: Principle of a client based OPC structure with one client

Fig. 9: : Principle of a client based OPC structure with two clients using a heartbeat signal

Another redundant application or client level approach uses at least a heartbeat signal as shown in Figure 9. In this approach, two clients are active and inform each other about its current health state, or the operating client activates the second client if a standby mode is preferred [32].

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

5 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 [4,17,24]. 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 [28,34]. 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 [4,5,28]. Therefore, chemical, pharmaceutical and process industries are also starting to look into the approach of collecting and evaluate vibrating,

rotating, thermal and other mechanical properties [28]. 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 is another reason, but mostly nobody is aware of the fact the IT equipment should be considered in the maintenance procedures [5,28]. Figure 10 shows a structure of IT maintenance, where computer devices, network devices and applications are also considered.
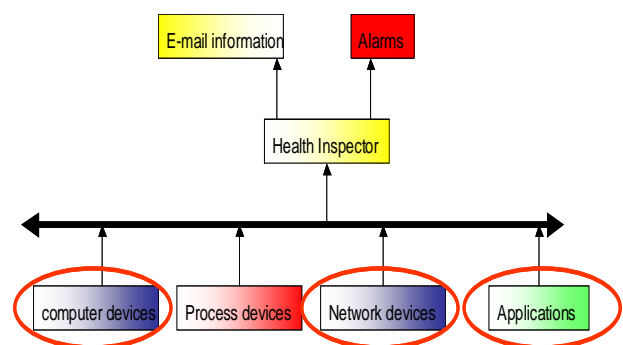

Fig. 10: Health inspector

Every device sends 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 sent 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 of each PC [28].

6 Process Condition Maintenance and Monitoring with OPC

This section will describe the necessary structures to implement and use a maintenance system, which is based on OPC client - sever approach.

6.1 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 [12,35,36]. 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 [12,34,35]. 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 [12]. 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 [12,35] .

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.
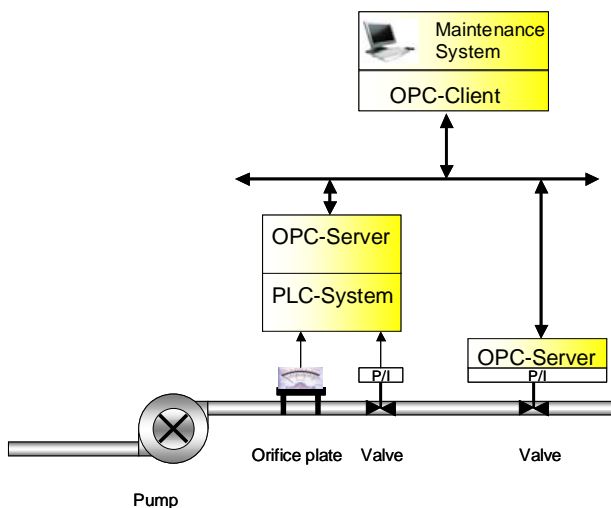

Fig. 11: Collecting data for maintenance system

Figure 11 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 an OPC server, or new and modern devices are already equipped with an OPC server to communicate with the client of the maintenance

system [12,34,36]. The advantage of new and modern sensors will be discussed within a separate section.

6.2 Condition based maintenance

After the communication and the capture of 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 [12]. 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.
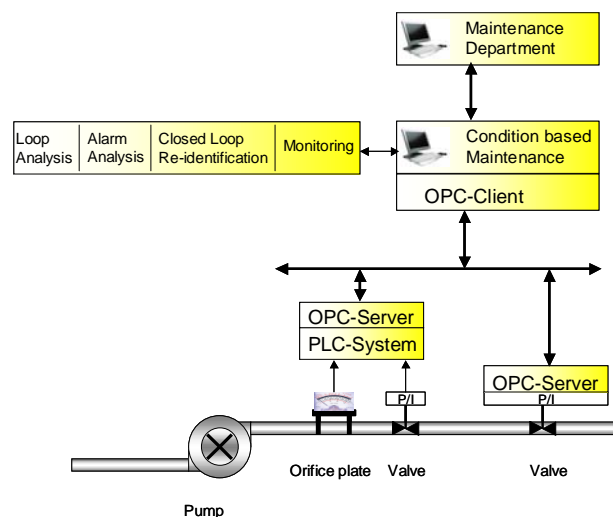

Fig. 12: Modern Maintenance System

One 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 [12].

Figure 12 sketches 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 [12].

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 [22], 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.

6.3 Identifying malfunctioning instruments

If an instrument breaks, then it is simple to identify it and repair or exchange it. But often, it is a process when an instrument starts to get malfunctioning, maybe it works in certain ranges correctly but in others not. So, it is very difficult for 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. This eliminates potential hazards, which might have resulted in unscheduled shutdowns.

6.4 Monitoring multiple factors

The advantage of modern maintenance system is that several variables and factors are 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. One example of a modern sensor presented here determines the degradation of lubricants in compressors or machines.

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 [1,19]. 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.

Modern on-line micro-sensor monitors the quality of lubricating oil in compressors or machines. Such sensors can exploit optical grating, micro-optical components and detection [1].

Other sensors estimate the health of compressors, machines or rotating elements by analysing the rotations and its vibrations or a rise of a temperature of devices can indicate that a device should be maintained or at least be monitored.

While it was necessary in the past to build the control- and analysing units onto the sensor, with an OPC- client-server architecture the sensors are only measuring the values and the analysis is done by the maintenance system. Therefore, complex software has not been squeezed onto a small micro-controller system.

6.5 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 [12,20] 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 [20,19]. Within in the last few years, process identification move towards closed-loop identification [20, 38]. 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. 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 to 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 Åström and Hägglund 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 [10], the control parameters can be estimated as demonstrated [2, 3, 20]. 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.

In describing function analysis it is assumed that only the fundamental harmonic component is significant. The assumption is often valid since most real systems have low pass filter characteristics that attenuate the higher harmonics. The describing function or the sinusoidal describing function is defined as the complex ratio of the fundamental harmonic component of the output of the non-linearity divided by the peak value of the sinusoidal input: appearing at the input to the non-linearity.

$$N = \frac{Y_1}{X} \angle \phi_1 \qquad (1)$$

Where N is the describing function, X defines the amplitude of the input sinusoid, Y_1 describes the amplitude of the harmonic component of the output and ϕ_1 is the phase shift of the fundamental harmonic component introduced by the non-linearity. The fundamental harmonic present in the non-linearity output is given by:

$$y_1(t) = \frac{4M}{\pi} \sin(wt) \qquad (2)$$

$$N = \frac{Y_1}{X} \angle 0° = \frac{4M}{\pi A} \qquad (3)$$

The output-height of a relay is M. The variable 'A' is the amplitude of the sinusoid at the input to the non-linearity. The link between the ultimate gain and describing function is expressed in Equation 4.

$$N(A) = \frac{4M}{\pi A} = K_u \qquad (4)$$

A schematic diagram of the relay setup is shown in Figure 13 and a typical system response is shown in Figure 14..
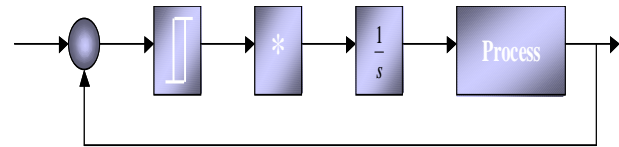

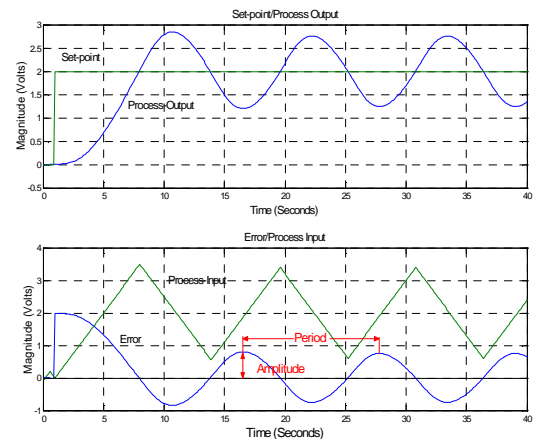Fig. 13: Åström–Hägglund controller tuning

Fig. 14 : Oscillated process and obtained data

Other identification methods to model a process with relay methods can be found in "Relay Feedback Analysis, Identification and Control" by Wang [38], 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.

7 Planned and Future Work

So far, the department of computer architecture and system programming at the University of Kassel has developed a fully functioning OPC client, but uses commercial OPC servers. Servers are normally the gateway of commercial products to communicate with other system items, equipment units or process elements and are mostly provided by the developing companies anyway. Therefore, it does not make a lot of sense to develop a server, just for the sake of doing it. The current development is therefore concentrating on the expansion of the client. The first step is going to be to develop a redundant client-structure which can operate with two or more servers simultaneously to receive and to process data and to compare it. The comparison procedure

evaluates the time stamp, process value and process quality and if one of the data-parameter-triples varies then an alarm is set to inform the operator or personnel. The different approaches have been introduced in this paper.

Since the system will be developed for industries, different guidelines, standards and procedures have to be followed, if a reliable software has to be implemented [4,6]. The scope of this paper was not to give guidelines on how software and hardware have to be implemented if standards as the IEC 61508 [7,14]should be applicable. Therefore, it is referred to *Design, Implementation and Testing of Safety-Related Software* [6] and *Functional Safety* [4,5].

Additionally, an investigation on how industrial communications can be made more reliable is an important issue [9,11,16,21,37], even if an client / server approach is used, which is operating above the actual communication.

So far, the new standard OPC-UA, has not been considered for our development, since it is too slow for real time applications, but since the development will move towards this standard, it is worthwhile to look into it.

8 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 such as control engineering and identification 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 maintenance system

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 does not leave this area to a few companies, but getting involved.

References:

[1] Adgar A., Schwarz M.H. & MacIntyre J. Development of Intelligent Software for a Micro-Sensor based Oil Quality Analysis System, *COMADOM 2004*, 2004, Cambridge, United Kingdom

[2] Åström K., and Hägglund T. *PID Controllers: Theory, Design and Tuning,* Instrument Society of America 67 Alexander Drive,PO Box 12277, Research Triangle Park, North Carolina 27709,USA, 1995

[3] Åström, K. & Hägglund, T., Automatic tuning of simple regulators with specifications on phase and amplitude margins, *Automatica*, Vol 20, 1984, pp 645-651.

[4] Börcsök J., *Elektronic Safety Systems*, Hüthig, 2004.

[5] Börcsök J., *Functional Safety*, Hüthig, 2007

[6] Börcsök J., Schäfer S., Design, Implementation and Testing of Safety-Related Software, *WSEAS Transactions on Systems and Control* Issue 2, Vol. 1, December. 2006, pp.149-154

[7] Börcsök J,. Schwarz M.H., An Investigation on how to Develop a Safety Related System in line with the Standard IEC 61508, *WSEAS Transactions on Systems and Control* Issue 2, Vol. 1, December. 2006, pp.300-305

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

[9] Frantti T., Reliable Simulation of Communication Networks, *WSEAS Transactions on Communications* Issue 5, Vol. 5, May. 2006, pp.758-765

[10] Gelb A. & Vander Velde, W.E., *Multiple-input describing functions and nonlinear system design*, McGraw-Hill, 1968.

[11] Gontean A., Otesteanu M., Rugina S., Sarbu-Doaga G., Versatile Communication Solution for PLC based Control Systems, *WSEAS Transactions on Communications* Issue 12, Vol. 5, December. 2006, pp.2137-2141

[12] Gould J. CMMS: Integrating Real-Time Information for Condition-Based Maintenance. *Matrikon, Inc*. 2005

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

[14] IEC 61508, International Standard 61508: *Functional safety of electrical/electronic/programmable electronic safety-related systems*, Geneva, International Electrotechnical Commission, 2000

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

[16] Kamikawa N., Umeo H., Sequence Generation Problem on Communication-Restricted Cellular Automata, *WSEAS Transactions on Communications* Issue 12, Vol. 5, December. 2006, pp.2219-2224

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

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

[19] Laaouad F., Hafaifa A. Laroussi K., A Nonlinear Model for a Turbo Compressor using Fuzzy Logic Approach. *WSEAS Transactions on Systems and Control* Issue 2, Vol. 2, February. 2007, pp.242-248

[20] Leva A., Cox C., Ruano *A. IFAC Professional Brief: Hands-on PID autotuning: a guide to better utilisation*, International Federation of Automatic Control

[21] Li J., Poulton G., James G., Zeman A., Wang P., Chadwick M., Piraveenan R., Performance of Multi-agent Cooerdination of distributed Energy Resources, *WSEAS Transactions on Systems and Control* Issue 1, Vol. 2, January. 2007, pp.52-58

[22] Li L., Li Q., Vibration Analysis Based on Full Multi-Body Model and the Active Control of Commercial Vehicle Suspension System, *WSEAS Transactions on Systems and Control* Issue 2, Vol. 2, February. 2007, pp.230-235

[23] Lin J., A Conceptual Model for Recognition and Management of Web-Services, *WSEAS Transactions on Communications* Issue 3, Vol. 6, March. 2007, pp.650-656

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

[25] Mitchell W. Historical OPC HAD: Helping Improve Plant Performance. *Matrikon, Inc*. 2003

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

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

[28] Murphy E. OPC Maintaining Healthy IT Assets. *Matrikon Inc*. 2006

[29] Nawaz T., Mian M. S., Habib H. A., TCP Enhancement: Fast Active Queue Management Scalable Transmission Control Protocol, *WSEAS Transactions on Communications* Issue 11, Vol. 5, November. 2006, pp.2100-2108

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

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

[32] 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

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

[34] Shin Daejung, Seung Y. Na, Jin Y. Kim, Seong-Joon Baek, Water Pollution Monitoring System by Automoous Fish Robots. *WSEAS Transactions on Systems and Control* Issue 1, Vol. 2, January. 2007, pp.32-38

[35] Slavov V., Tashev T., Data Acquisition Units for Student Labs, *WSEAS Transactions on Systems and Control* Issue 2, Vol. 1, December. 2006, pp.332-335

[36] Torabian H. Mohammadion A., Design and Construction of Non-Contact Data Acquisition System for Robot Performance Measurement, *WSEAS Transactions on Systems and Control* Issue 2, Vol. 2, February. 2007, pp.236-241

[37] Tran D. T., Choi E., High Performance Communication of a Reliable User Datagram Protocol, *WSEAS Transactions on Communications* Issue 3, Vol. 6, March. 2007, pp.464-479

[38] Wang, Quing-Guo, Lee, Tong H., Chong, Lin, *Relay Feedback Analysis, Identification and Control*, Springer 2003