An Open Communication System for remote control and monitoring dedicated to Industrials Applications over TCP-IP applied to public networks

Núnzio Torrisi[†], Fábio Ferraz Jr.[††], Carlos M. Valente[††], Arthur P. S. Braga[††], João F. G. Oliveira[††]

[†]*Dipartimento di Ingegneria Informatica e delle Telecomunicazioni University of Catania.*
V.le A. Doria, 6 - I9510, Catania, Italy

[††]*Departamento de Engenharia de Produção, Universidade de São Paulo.*
Av. Trabalhador Sãocarlense, 400, São Carlos, SP, 13566-590, Brasil

E-mails: `ntorrisi@diit.unict.it; {fferrazj,cmov,abraga,jfgo}@sc.usp.br`

Resumo— Durante os últimos anos, a evolução das redes fieldbus, em conjunto com a simplificação das redes tradicionais de computadores, permitiu a integração de diferentes sistemas de comunicação tanto dentro das células de produção, quanto entre células de produção e outros sistemas para gerência inteligente, supervisão e controle. Várias tecnologias de comunicação já existem atualmente para redes públicas e não-públicas como a Internet e as Intranets. Mas uma vez que a maioria das aplicações industriais é *time-critical*, as exigências dos sistemas de comunicação para controle remoto são diferentes daquelas em aplicações comuns de redes de computadores conectadas à Internet como Web, e-mail e transferência de arquivo. Além disso, juntamente com as exigências de tráfego sensível a atrasos, as exigências de segurança de dados devem ser consideradas quando o controle remoto é realizado sobre redes públicas como a Internet. A solução de comunicação apresentada neste trabalho é denominado projeto CyberOPC. Este inclui o estudo e a realização de um novo protocolo aberto para o controle remoto de sistemas industriais, os quais em nosso caso são máquinas CNC, satisfazendo as exigências de tempo e segurança. O trabalho descrito, em síntese, utiliza metodologias já famosas e novas abordagens de redes de computadores e Segurança cibernética no desenvolvimento de um novo protocolo de transporte para os dados do processo com os seguintes objetivos: **(i) minimizar os atrasos na transmissão de dados de controle sensíveis a atrasos; (ii) garantir a segurança do canal de comunicação utilizado; (iii) garantir a integridade e segredo dos dados de controle transmitidos.**

Palavras-chave– Monitoramento e Controle Remoto, OPC, rede industrial, protocolo de rede.

Abstract— During these last years the evolution of the fieldbus networks together with traditional computers networks have simplified and allowed the integration of the different communication systems within both the single cells of production and between production cells and other systems for business intelligence, supervision and control. Various communication technologies already adopted exist today for public and non-public networks as Internet and Intranet. Since the most of the industrial applications are time-critical, the requirements of communication systems for remote control are different from the common applications for computer networks used in Internet like Web, e-mail and file transfer. Moreover, together with the requirements for time-critical traffic, the requirements of cyber security must be considered when the remote control is built over public networks like Internet. The communication solution, outlined in this work is proposed as the called CyberOPC Project. It includes the study and the realization of a new open protocol for the remote control of industrial systems, which in our study case are machinery CNC, satisfying the time-critical and cyber security requirements. The described work, in synthesis, will arrange already famous methodologies and new approaches of the Computer Networks and Cyber Security in order to develop a new protocol of transport for data process with the following goals: **(i) to minimize the delays of transmission for time-critical control data; (ii) to guarantee the security of the used communication channel; (iii) to guarantee integrity and confidentiality of the transmitted control data.**

Keywords— Remote Control and Monitoring, OPC, industrial network, network protocol.

1. Introduction

The choice of the communication systems adopted inside manufacturing enterprises influences all the production strategies (Bangemann, Hahniche and Neumann, 1998). The main issues are related to intelligent data collection, remote controls, and integration of remote processes. For such reason, the distributed networks for manufacturing have been widely studied in the last 20 years up to the development of the fieldbus like standard technologies for communications in process control (Georgoudakis et alli, 2003). Besides, Ethernet prominence was catalytic for the developments in the area of Fieldbus standards. As we heading towards an all-IP world, the battle for supremacy in the industrial networks field has now been transferred in the upper layers of the TCP/IP stack as opposed to the wars fought all the way down to the physical medium. The major players in this battleground are OPC Foundation (OPC, 2006) which promotes OPC (OLE for Process Control), and IDA (Interface for Distributed Automation) which supports RTPS (Real Time Streaming Protocol), ModBus and Foundation Fieldbus HSE. On the machinery control side can be found quite a few competing protocols such as ProfiNet, EtherNet/IP, DeviceNet and others (Polsonetti, 2002).

The various communications solutions today adopted keep the historical problem that the data of different systems have different formats and different communication protocols. This is very important when, for example, drives are connected to a PC-based SCADA (Supervisory Control and Data Acquisition) system. Software vendors, with their process monitoring, control, and data management system were required to develop an individual I/O driver for each protocol. Clearly the development, by software vendors, of unique drivers for each different type of plant floor control equipment is not only time consuming, and inefficient, but also inherently adds additional risks to the successful and timely completion of a project. For these reasons, five companies, with Microsoft®, decided to work together to develop the OPC technology (OPC, 2006).

Today, the OPC technologies combined with Web technologies, as WebServices, XML and SOAP (Simple Object Access Protocol), are used to draw complex architecture for manufacturing in order to create communication system directly between shop-floor and decentralized supervision systems (Zheng and Nakagawa, 2002). The OPC technology, which today is a standard for the monitoring and control in factory floor (Pattle and Ramisch, 1997), does not offer solutions for communication over the Internet, with jitters under one second.

This work proposes a new open protocol for the remote control and monitoring of the industrial systems, which in our case are machinery CNC, satisfying the time-critical and cyber security requirements (Ferraz Jr and Coelho, 2005; Oliveira et alli, 2002). The network for the experimentation of the protocols and the methodologies applied to remote process control requires high transfer rate with bandwidth reserved. The Kyatera (2006) network of the FAPESP/TIDIA Program (Tecnologia da Informação no Desenvolvimento da Internet Avançada) fits the requirements for speed and bandwidth management.

The distribution of this paper is organized as follows: Section 2 discusses the main problems related to the distributed process, and the most relevant cyber-security problems related to communications over public networks; Section 3 exposes a short introduction about the OPC technology; Section 4 shows an overview of the CyberOPC communication system, and presents the proposed architecture; Section 5 exposes the initial tests done inside the NUMA laboratory; and the conclusions are discussed in Section 6.

2. Communication problems related to remote control and monitoring of distributed control processes

In Distributed Process Control Systems, DPCSs, although the flow of information produced by the processes, typical of process control environments, may differ greatly according to the specific plant being considered, it can generally be divided into – synchronous (generated by periodical processes that carry out actions repeated with a constant frequency) and asynchronous (produced by processes that evolve in time in a way that cannot be foreseen a priori) flows. Besides the problem of this difference in the nature of the information flow, the sophisticated architectures of today's information systems feature a large number of points of attack: information is routed through a wide range of stations on its way from the source address to the destination address, including mobile workstations, field offices, network nodes and sub networks.

An example of **Synchronous Flow** is a sampling process that receives an analog signal from a sensor and produces a digital signal at a frequency equal to that of sampling. Each periodically produced data item is consumed by one or more consumer processes, which are also periodic processes. Data has a lifetime that corresponds to the interval between the generation of two consecutive samples, as shown below in the Fig 1.

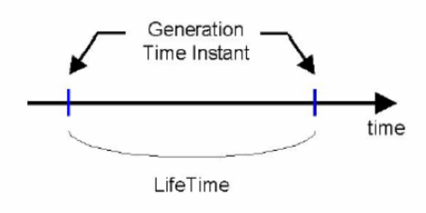

Figure 1 - Lifetime of periodic traffic

An example of this is a consumer process reconstructing a transmitted digital signal to obtain the original analog signal. A consumer process consuming a periodically updated variable generally has to receive each value within a maximum admissible interval that corresponds to the lifetime of the data (Figure 2).

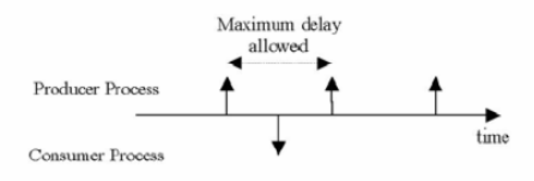

Figure 2 - Time constraints of producer and consumer processes

The transmission of periodic data should, in theory, occur at fixed time instants; however, the real transmission instant is always located somewhere around the ideal instant, which is known as jitter. For a consumer process reconstructing an original digital signal, the maximum jitter allowed is represented by the sampling interval (Figure 3).

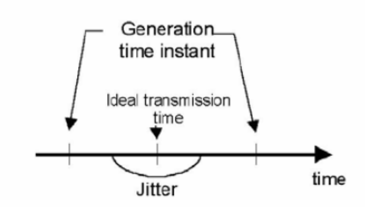

Figure 3 - Jitter during the transmission of periodic data

As a periodic information flow is linked to processes with known dynamics, the production and consumption periods and the jitter are known a priori. However, there are processes where the information flow is **Asynchronous** - this term asynchronous is used to highlight the absolute independence between activity on the communication channel (marked by a system clock) and that of the processes producing the traffic. In an industrial process control system there are numerous examples of asynchronous flows, for example alarms generated by a supervision system, operations to *download* intelligent *controller* configurations, or queries to centralized or distributed databases.

The CyberOPC communication system is our proposal to overcome the above-mentioned problems to remote control and monitoring of industrial systems. The next two subsections overviews the proposed communication system, and its architecture, respectively.

3. Integration, Supervision and Control using the OPC technology

OPC (2006) is a set of interfaces defined by the OPC Foundation, initially based on OLE/COM (Object Linking and Embedding/Component Object Model) and DCOM (Distributed Component Object Model) technology, for open software application interoperability between automation/control applications, field systems/devices and business/office applications. The basic principle of OPC operation is that an OPC client, as a SCADA software for example, transfers data to/from PLCs/Devices by means of an OPC server (see Fig. 4). The OPC client can operate either locally (server on same PC) or via local network, and the server accesses PLCs/Devices via drivers (e.g. drivers to fieldbus and Ethernet based products).

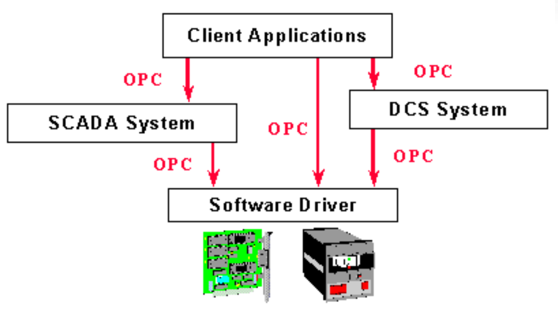

Figure 4 - General OPC Communications diagram

OPC is open connectivity in industrial automation and the enterprise systems that support industry. Interoperability is assured through the creation and maintenance of open standards specifications. The core of the OPC paradigm resides in its single database furnishing data: the OPC servers themselves are the database. Standard DCOM provides instant access to both clients and servers from any node. OPC this way became the universal connectivity standard for manufacturing and factory/plant floor devices and systems (Hong et alli, 2002), allowing many devices and applications to be tied together from multiple vendors, thanks to its standard interface based upon COM/DCOM. To extend the use of OPC over TCP-IP public networks, the OPC Foundation suggests adopting Web services technologies. One great innovation of the current project comes that the OPC Foundation does not investigate the important correlation between the network quality of service and the time critical requirements for process control currently – a subject in this proposal.

4. The CyberOPC communication system

The proposed protocol, in synthesis, will arrange already famous methodologies and new approaches of Computer Networks and Cyber Security in order to develop a new protocol for the transport of process data with the following goals:

(i) To diminish the transmission delays for time-critical data;

(ii) To guarantee the security of the communication channel used;

(iii) To guarantee the integrity and confidentiality of the transmitted messages.

To monitor and to control industrial processes means to understand the related dynamics of the processes and regularities of the variables. Analyzing the jitter, the life cycles and the regularities with which the processes use the cyclic variables, we can calculate the temporal requirements for the communication system used in the remote control.

In our case, the industrial processes for the grinding cycles made by CNC machineries, impose tight temporal constraints on the variable to control. Some initial experimental tests (see Section 5), already done inside of laboratory NUMA (Núcleo de Manufatura Avançada), have pointed out the difficulties to control and monitor remotely a complete cycle of grinding also having an high speed network but without guaranteed bandwidth - which is supported in the KyaTera network. With the bandwidth reserved managed on the KyaTera network, it will be possible to define also various QoS (Quality of Services) for the remote control and monitor service. In addition to the problem of time-critical communication system, we must also consider the security problems related to the use of public networks as Internet.

The necessity to satisfy the time-critical and security requirements for the remote control has pushed to the study of a new protocol for process control. To obtain the maximum interoperability with the already existing factory floor technologies, we will construct our communication project over the OPC technology (Ling, Chen and Yu, 2004; Ding et alli, 2003).

Moreover, today a standard version of OPC technology for the web is not available; there are only proprietary solutions (Torrisi, Mirabella and Bello, 2005). We will call CyberOPC the new protocol that we propose here in order to develop a suitable solution and to test it on the KyaTera network.

4.1. Architecture of the CyberOPC Gateway

The proposed CyberOPC communication system previews the use of a gateway station called

CyberOPC gateway that will process messages sent to OPC towards the public network and vice versa. About the QoS, the communication system proposed is targeted for best effort network with a minimum of bandwidth reserved for periodic traffic.

Unlike many OPC gateways for Internet (Kapsalis et alli, 2003) that use WebServices with HTTP and SOAP, the CyberOPC gateway does not use WebServices because the performance loss will not be balanced by the advantage derived from the high level programming offered from the WebServices. However, in order to make use of WebServices that transports OPC data, OPC libraries for the processing of OPC messages are required. Therefore, assuming necessary the use of libraries, which are not open source and generally not free, for the processing of OPC data, we believe it will be useful to develop a set of free and open source libraries in order to obtain OPC communications on Internet with the best possible performances.

We begin therefore to analyze one OPC architecture gateway based on WebServices with HTTP and SOAP as shown in Figure 5.

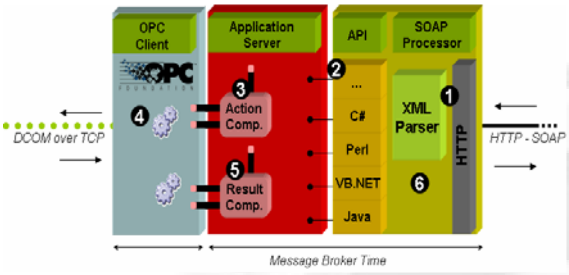

Figure 5 - Message Broker Time for WebServices OPC Gateway

The architecture shown in Figure 5 outlines in blocks the various processing steps for a generic message received from Internet. These steps are necessary in order to retrieve one or more OPC data. A request is managed by the remote Internet client through a call to the relative WebServices that will receive the request, and will generate the inherent HTTP-SOAP message directed to the gateway. Each gateway has to include a SOAP processor in order to interpret the SOAP part of the messages (step 1 of fig.5). After the SOAP processing, the gateway can call the related API (step 2 of fig.5) to manage the right OPC functions called by the Application Server (steps 3 and 4 of fig.5). The data results of the OPC functions will be reprocessed for first from the Application Server (step 5 of fig.5) and after from SOAP processor (step 6 of fig.5) before to send the response.

Different from the above architecture, the proposed CyberOPC gateway computes directly from the URL a set of OPC-XML-DA (Ling and Yu, 2002) complaints commands:

- **GetStatus:** It provides a common mechanism for checking the status of the server.
- **Read:** It provides the ability to read the value and quality for one or more OPC items. Other attributes, such as timestamp, can optionally be requested for items.
- **GetProperties:** It retrieves the properties of a selected OPC Item Path.
- **Subscribe:** The client application initiates the subscription and agrees to issue periodic refresh requests. This mechanism can be used to reduce the latency time of reporting a value change to a client and minimize the number of round trips between the client and server.
- **SubscriptionPolledRefresh:** Refreshes the data items from the last *SubscriptionPolledRefresh*.
- **Write:** This service writes the value for one or more OPC items.
- **Browse:** The server will do a *Browse* from the level specified by the combination of OPC ItemPath and ItemName.
- **SubscriptionCancel:** The server will cancel a subscription and allow the server to clean up any resources associated with the subscription. The server will cancel any processing in progress associated with the specified subscription.

So, there are only 8 OPC possible commands mapped into the URL. For this reason the "Command Parser", that has the role to recognize these commands, is simpler than any XML parser for SOAP messages (Füricht et alli, 2002)

The OPC commands are executed quickly and, in the case of periodic data request, we could enhance the response time using a dedicated OPC cache shared by the OPC Client and the http broker. A fast OPC data cache could be wrote asynchronously by the OPC Client for all periodic data request by the Internet Remote Client (Figure 6).

The step 1 in the figure 6 represents the processing of the CyberOPC commands without SOAP preprocessor. The introduction of the OPC cache strongly reduces the calls to the OPC Client. Some initial tests, detailed in the Section 5, have reported the reduction of 70% of the Message Broker Time if compared with the time consumed by a Gateway WebServices based. The steps 2, 3 and 4 represent the interaction between the OPC library and the CyberOPC Application Server.

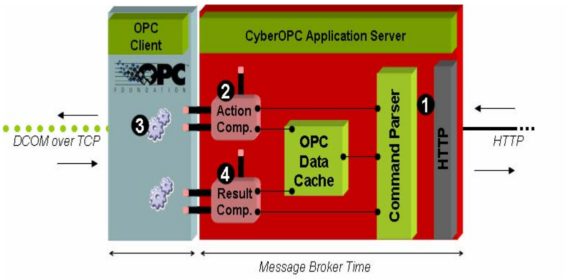

Figure 6 - Message Broker Time for CyberOPC Gateway for OPC

Internet communication typically runs through multiple program layers on a server before getting to the requested data such as a web page or a webservice. The outer layer is the first to be hit by the request. This is the high level protocols such as HTTP (web server), IMAP (mail server), and FTP (file transfer). Determining which outer layer protocol will handle the request depends on the type of request made by the client. In order to guarantee the integrity and confidentiality of the transmitted messages, the CyberOPC communication system adopts the HTTP high-level protocol then processes the request through the Secure Sockets Layer.

5. Initial Results Obtained

Our test bench for remote control of grinding process, inside the NUMA laboratory, was made up of (Figure 7):

- A ZEMA CNC connected to the NUMA network using a GE Fanuc Focas1 card;
- A sensors system installed inside the cabin of the ZEMA CNC and linked to a DAQ card installed on the HMI station; the HMI station was equipped with an OPC Server with the driver for GE Fanuc Focas1 and the DAQ card;
- A station with a public interface with the role of gateway for the access to the industrial data. This is equipped with an OPC software compliant and WebServices enabled using SOAP over HTTP. In the same station we have installed also our first simplified prototype of CyberOPC gateway;
- A remote Internet client out of the São Carlos USP network equipped with the CyberOPC communication libraries.

The remote control tests of grinding cycle ran on the ZEMA. The results show the performance difference between the five communications systems adopted from the production cell to the remote station.

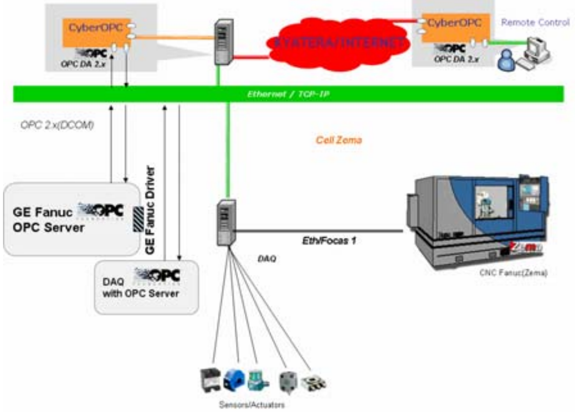

Figure 7 - NUMA-OPF Architecture for testing on KyaTera

Figure 8 shows the performance of different communication architectures in terms of minimum time to send and receive data. In particular, from the analysis of the traffic recorded during the tests, the following performance differences between the adopted technologies of communication are emerged in Table 1, where we summarize the round trip time and the periodicity average for each communication system.

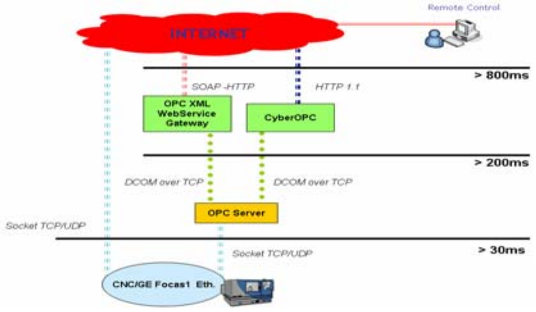

Figure 8 - Data transmission periodicity

The traffic labeled as **"Focas – Intranet Client"** in Table 1 represents the data flow exchanged directly between the control unit of the ZEMA CNC and a remote PC installed out of the NUMA Intranet. This data flow is carried using socket technology on TCP and UDP. For both communications endpoints, the libraries to manage the GE Fanuc data are required.

Table 1 - Comparison Table in milliseconds

	Periodicity	***RoundTrip Time***
Focas – Intranet Client	50,000	1,500
Focas – Intranet OPC Server	62	11,1
Intranet OPC Server – CyberOPC GW and WebService GW	250	30
Internet Remote Client – WebServices GW	1100	40
Internet Remote Client – CyberOPC GW	800	40

The GE Fanuc communication system uses the TCP port 8192 and UDP port 8191 with binary encoding. For such reason they are for default cut outside from the security policy of most routers and firewalls since the UDP traffic, incoming from not standard and well known services, is considered, generally, incoming traffic from malicious software . These kinds of communications are so called not firewall-friendly.

The traffic labeled as **"Focas – Intranet OPC Server"** represents data flow exchanged directly from the control unit of the ZEMA CNC and the OPC Server installed on HMI station. Also for this kind of communication for both endpoints is needed to use the libraries to manage the GE Fanuc data.

The traffic labeled as **"Intranet OPC Server – CyberOPC GW – WebServices GW"** represents data flow exchanged between the OPC Server, installed on the HMI station, and the CyberOPC and WebServices gateways. Both use, for all OPC communication in the Intranet, the same OPC Client library with the same network and data configuration. The gateways differ for the communication approaches used for Internet.

The traffic labeled as **"Internet Remote Client – WebServices GW"** represents data flow exchanged between the WebServices gateway and a remote PC station with the WebServices library to communicate using HTTP-SOAP.

The traffic labeled as **"Internet Remote Client – CyberOPC GW"** represents data flow exchanged between the CyberOPC gateway and a remote Internet PC station equipped with the CyberOPC library to send OPC commands over HTTP.

Both gateways, adopting the HTTP protocol, are firewall-friendly and could be possible to achieve the security problems for the channel, by integrating the use of digital certificates and HTTPS (Freier, Karlton and Kocher, 1996). The important performance difference between the CyberOPC and WebServices communications, that justify and encourage our future works, is that the minimal time to schedule, process and send a message using the WebServices is more than one second while the same time for the CyberOPC is not more of 800 milliseconds.

In order to analyze the performance of the CyberOPC communication system over internet to control time critical processes, as grinding processes, we have to focalize our attention to the round trip time of the communication system. This because the round trip time is strictly related to the refresh time of the periodical data processes. Our basic CyberOPC communication system provides a round trip time in the average enough to support a refresh time for data process around 800-900 ms.

6. Conclusions

The important performance difference between the proposed CyberOPC and WebServices communications, that justify and encourage our future works, is that the minimal time to schedule, process and send a message using the WebServices demands more than one second, while the same time for the CyberOPC is not more than 800 milliseconds. Using the technology supported by the KyaTera (2006) network, we can also suppose to improve the CyberOPC prototype in order to obtain a periodic data processing with about 500 milliseconds.

Acknowledgments

The authors would like to thank the support from FAPESP, CNPq, Zema Industrial Grinders and GE Fanuc who provided financial support and donation of the equipment.

References

Bangemann, T., Hahniche, J., and Neumann, P. (1998). Integration of fieldbus systems in computer aided facility, *Proceedings of the IECON '98, vol.3, pp. 1835-1840.*

Ding, Z. Q.; Aendenroomer, A.; He, H. and Goh, K. M. (2003). OPC based device management and communication in a distributed control application platform, *Proceedings of INDIN 2003, pp. 107-111.*

Ferraz Jr, F. and Coelho, R.T. (2005). Data acquisition and Monitoring Tools with CNC of Open Architecture Using Internet, *The International Journal of Advanced Manufacturing Technology*, V. 26, pp. 90-97.

Freier, A. O.; Karlton, P. and Kocher, P. C. (1996). The SSL Protocol Version 3.0, *Available: http://wp.netscape.com/eng/ssl3/ssl-toc.html.*

Füricht, R.; Prähofer, H.; Hofinger, T. and Altmann, J. (2002). A component-based application framework for manufacturing execution systems in C# and .NET. *Proceedings of the CRPITS '02, Australian Computer Society, Inc., pp. 169-178.*

Georgoudakis, M.; Kapsalis, V.; Koubias, S. and Papadopoulos, G. (2003). Advancements, trends and real-time considerations in industrial Ethernet protocols, *Proceedings of the INDIN 2003, pp. 112-117.*

Hong, X.; Jianhua, W.; GuiQing, Z. and ShiQuan, Z. (2002). Using data engine in distributed automation system, *Proceedings of the PowerCon 2002, vol.3, pp. 1744-1747.*

Kapsalis, V.; Charatsis, K.; Georgoudakis, M. and Papadopoulos, G. (2003). Architecture for Web-based services integration, *Proceedings of the IECON '03, vol. 1, pp. 866-871.*

KyaTera – FAPESP. (2006). *Available: www.kyatera.fapesp.br.*

Ling, Z. and Yu, J. (2002). The design of SCADA based on industrial Ethernet, *Proceedings of the 4th World Congress on Intelligent Control and Automation, vol.4, pp. 2786-2789.*

Ling, Z.; Chen, W and Yu, J. (2004). Research and implementation of OPC server based on data access specification, *Proceedings of the WCICA 2004. vol. 2, pp. 1475-1478.*

Pattle, R. and Ramisch, J. (1997). OPC the de facto standard for real time communication, *Proceedings of the Joint Workshop on Parallel and Distributed Real-Time Systems, pp. 289-294.*

Polsonetti, C. (2002). Industrial Ethernet protocols: the next battleground?, *The Industrial Ethernet Book, Issue 12, Available: http://ethernet.industrial-networking.com/articles/articledisplay.asp?id=26.*

Oliveira, J. F. G.; Silva, E. J.; Biffi, M. and Matrai, F. (2002). Open Architecture Control System For High Speed Grinding. *Imts 2002 – SME - Manufacturing Conference, Chicago*.

OPC Foundation Website. (2006). *Available: http://www.opcfoundation.org/.*

Torrisi, N; Mirabella, O. and Bello, L. L. (2005). A General Approach to model traceability systems in food manufacturing chains, *Proceedings of IEEE ETFA05, Catania.*

Zheng, L. and Nakagawa, H. (2002). OPC (OLE for process control) specification and its developments, *SICE 2002. Proceedings of the 41st SICE Annual Conference*, pp. 917-920 vol.2.