Some Problems of Integrating Industrial Network Control Systems Using Service Oriented Architecture

Marcin Fojcik and Joar Sande

Sogn og Fjordane University College
{marcin.fojcik,joar.sande}@hisf.no

Abstract. In this paper, the authors present methods for connection Service Oriented Architecture with OPC UA to control systems. Different disciplines such as data science, communication theory and control systems have different viewpoint on the properties of communication systems. Requirements of the control systems and requirements of information systems are presented. The integration of the two systems is analysed. Problem is formulated: how it is possible to get new functionality without losing existing features. Most important parameter is time. The control systems have to achieve fixed time of data sampling, regardless of situation. For Service Oriented Architecture time is not so important. SOA emphasizes openness, modularity and compatibility. Integration of two different systems and integration problems in distributed control systems are presented and analysed.

Keywords: OPC Unified Architecture, Network Control Systems, Service Oriented Architecture, fieldbus, network traffic.

1 Introduction

Today, information systems are designed more open and modular. The information systems consist of universal sets of interfaces and services. It simplifies creation, installation and maintenance. The reason for this is the use of one communication protocol – TCP/IP, as a standard. Standardization enables easy connecting between different applications. Integration gives possibility of using the same services, common data exchange and independence of operating systems and hardware. Similar, unified tendency is observable in automation systems. Unfortunately, one common communication standard is not available here. There are many different, incompatible standards define to data exchange in completely different situations and environments. Connection of some of these standard communication protocols in one system is possible. This action requires individual processing, human intervention is often necessary.

There are many research projects [1,2,3] with aim to define universal rules and methods for integration. The aim is to achieve an open control system. The pursuit on openness can lead to loss the characteristic properties of the

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2013, CCIS 370, pp. 210–221, 2013.

system. In industry protocols, typically the raw data without description are sent. Without knowledge about the system and protocol configuration, these data are not usable. Adding descriptions to the protocol increases the amount of data to transfer, and therefore increases the transfer time. Other, potentially negative, situation is the mutual influence of interfaces for the control system and the information system. Both have different communication possibilities: fast transfer of small amount of data versus slow transfer of big amount of data. It is possible to send all information from the control system into the information system. Transfer in the other direction leads to increased traffic and blockage of the control system.

In this paper, the authors suggest methods to connect information systems based on Service Oriented Architecture to industrial networks. Some problems of integrations are also shown. Requirements, especially for data transfer speed, are described. Authors present description and analysis of some control system protocols and services available in Service Oriented Architecture. In particular, services used to connection SOA to the control system are considered. With analysis of existing research projects, literature and own research, authors show requirements necessary to correct integration. Nowadays, software and hardware do not allow full integration, but even partial integration gives many new possibilities to reduce time of creating software for automation systems.

This paper is divided into 6 parts. In Section 2, the authors present time problems in control systems; Sect. 3 contains mathematical equations used during parametrization of industry network. In Section 4 the ideas of the SOA based system are presented. Section 5 contains description of the OPC UA protocol and shows how it can be used to connect SOA and industry network. In Section 6, results of calculations and experiments in connecting are presented.

2 Control Systems

In a control system there are several control loops, some of them single loop, others with multiple inputs and outputs. Some of the control loops can be interconnected such that change in a control parameter in one control loop affects a different loop. The control loop calculates new parameters based on measurements from the field, which in turn are sent to an actuator. Some loops e.g. temperature control, can be slow, while others e.g. motor control, requires an immediate response. In some cases the states of a control system can be estimated based on inputs, outputs and the physical measurements that are done. This lowers the costs and the number of sensors can be minimized, without reducing the quality of the control.

A process can be described by a state space model, where x are the states, u is the input and y it the output.

$$x' = A * x + B * u \,, \qquad y = C * x \,, \qquad \text{for continuous control,}$$
$$x_{k+1} = \Phi * x_k + \Gamma * u_k \,, \qquad y(k) = C * x(k), \qquad \text{for discrete control,}$$

where:

- $\Phi = e^{Ah}$,
- $\Gamma = B * \int_0^h e^{As} \mathrm{d}s$,
- A, B and C are matrices describing the controlled system: A – system matrix, B – input matrix, C – output matrix,
- h is fixed sampling period.

Figure 1 shows a control schema of a process plant [4]. The measurements $y(t)$ are sampled at time t_k. A mathematical model calculates the different states of the control system. The states are combined into an output from the controller(s) to the actuator(s), controlling different plant processes involving e.g. temperature and motor speed. The sampling time t_k is crucial, it must be short enough to detect any changes in the controlled process. A slow sampling rate can lead to an unstable system. The controller will overreact; the controlled system will receive too much energy, and therefore becomes unstable. The destruction of both equipment and production batch can be the result. Communication through a network induces delays. The duration between sampling of measurement signal until it is used by the actuator, introduces an additional time delays that must be dealt with. Time delay caused by the transmission of measurements to the controller, processing and then sending data from the controller (Fig. 1) is defined [4] as:

$$t_{\text{delay}} = t_1 + t_2 + t_3 \quad .$$

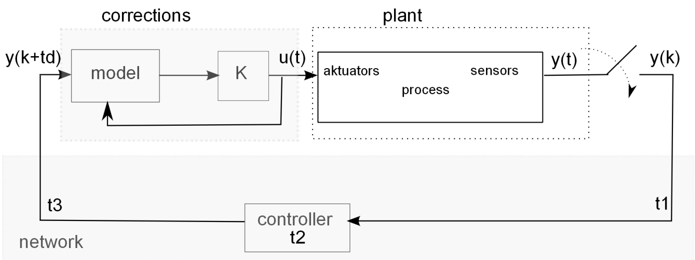

Fig. 1. Model of control system with network delays

In order to maintain correct operation of the control system, time delay tdelay should be equal or lower than fixed sampling period h.

$$t_{\text{delay}} \leq h \quad . \tag{1}$$

Some cases of control systems with non-constant time delay can be corrected with additional mechanisms. One of them is Smith Predictor – a delay compensation algorithm. Other strategies are the Dahlin Algorithm [5] or the Kalman filter [6] which can be used to refine the output signal from imprecise input data but with reduced quality.

In [5] response time and computation time for the cases of control systems without delays are presented (Table 1). Changing of the time delay leads to serious (up to 46 times longer) extension of the response time.

Table 1. Comparison of some correction strategies for control systems

	Control system		Control + Smith Predictor		Control + Dahlin algorithm	
	computation time [ms]	response time [ms]	computation time [ms]	response time [ms]	computation time [ms]	response time [ms]
nodelay	20	2.4	40	5	40	240
delay without computation time stability	—	—	—	230	—	480
with controller optimization	—	—	40	150	40	350

3 Data Exchange in Industry Networks

For Network Control System, the performance is a function of not only the sampling period, but also the traffic load on the network [7]. The traffic load can cause variable time delays in the transmission of data to the control system. The control system requires that time delays should not be longer than specified sampling time (1) It is necessary to calculate maximum time delay for known configuration of the control system.

Several have calculated the transmission time used to send one data package. We refer to [8,9] for such calculations. This approach is often insufficient in distributed control systems, because of protocols requirements. To achieve proper control, continuously sending of many measurements and control signals in specified time is necessary. For example, in protocols based on pooling, a slave station can send data after receiving a request from the Master station, and then wait for the next request, while the Master station asks the next Slave station. To calculate the shortest time period between two transmissions of the same data, we need to know the structure of the entire distributed system (protocol model, amount of devices, processing times for each device, transmission speed, and many others). The time used to send the data between all devices is called network cycle [10]. This time can be calculated for all protocols, which guarantee constant and repeatable actions.

The traffic depends on the network protocol and the configuration. There are many protocols according to different needs. Industrial protocols based on some models: Time Division Multiple Access, token, Master-Slave or Producer-Distributor-Consumer, have precisely defined order of data exchanges. Data exchanges are executed strictly in accordance to pre-defined scenarios. This guarantees receiving all data within a fixed, calculated time. Dense traffic caused by continuous data exchange is a disadvantage in these models (Fig. 2a).

Another solution is to send data on demand only. Devices, with additional processor, send data only on events. Protocols based on this model induce much

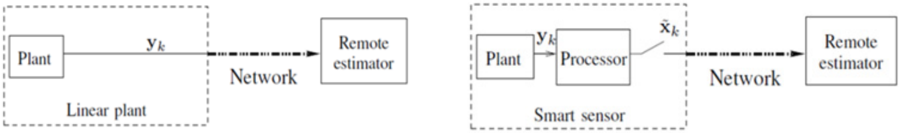

Fig. 2. Types of industry networks: (a) with constant network cycle, (b) event-driven [11]

lower network traffic, but it is not possible to plan the order of exchanges, it is therefore not possible to get fixed frequency of data exchanges (Fig. 2b). The first type of protocols requires calculation and planning of time parameters in the installation phase. In the working phase data packages are sent according to a planned scenario. Due to exchange-control mechanisms, these types of protocols guarantee regular delivery of the data. Data are sent continuously in the same time interval, independent of the actual situation. To achieve required network cycle time, the protocol generates a significant traffic on the network. Connection of a new device to an existing system, without renewing the configuration can easily lead to traffic blockage.

Event-driven data exchanges are used in protocols based on random or hierarchic access to media. Every device has a processor which chooses to send data or not. Only measurements which have changed value are transmitted. Operation of choosing data before sending reduces the network traffic. It is possible to detect, transmit and receive all, even short-term changes in process. These events cannot be detected in protocols with constant cycle time. A drawback of such protocols is that during an alarm situation many sensors suddenly detect changed values. A cascade occurs when several sensors send data almost at the same time. Such cascades can result in a blocked network. As an example, random access to media with collision detection or avoidance (CSMA/CD, CSMA/CA) can be mentioned [8,9,12]. A possible solution is to use data packets with priority. Information with lower priority has to wait until all information with higher priority is transmitted. This mechanism is used in Producer-Consumer (CAN) [13].

The network cycle time in protocols with constant time of cycle can be calculated. To simplify the problem, we assume only correct hardware condition for devices. For TDMA, the time is available as a protocol parameter. For calculations for protocols with *token* it is necessary to know the amount of devices, amount of exchanges and how the data are sampled, processed, prepared and sent by each device. In [10,14,15] the cycle time is calculated for many industrial protocols.

Minimal network cycle time for *token* protocols can be calculated by the formula:

$$T_{\mathrm{NCT}} = time_{\text{token transmission}} + time_{\text{processing}} + time_{\text{data transmission}} + \\ + time_{\text{data detection}} + time_{\text{acknowledgement transmission}}$$

$$T_{\mathrm{NCT}} = 3 * L_{\mathrm{A}} * (T_{\mathrm{TR}} + T_{\mathrm{DR}} + T_{\mathrm{AR}} + T_{\mathrm{PR}}) + \sum_{i=1}^{L_{\mathrm{A}}} T_{\mathrm{A_i}} + \\ + \sum_{i=1}^{L_{\mathrm{A}}} (T_{\mathrm{PR}_i} + T_{\mathrm{TR}_i}) + 2 * L_{\mathrm{A}} * (T_{\mathrm{DR}} + T_{\mathrm{AR}}) + L_{\mathrm{A}} * T_{\mathrm{TP}} \quad (2)$$

Respectively, calculation for Master-Slave protocols:

$$T_{\mathrm{NCT}} = time_{\text{request transmission}} + time_{\text{data detection and processing}} + \\ + time_{\text{answer transmission}}$$

$$T_{\mathrm{NCT}} = \sum_{i=1}^{L_{\mathrm{A}}} (T_{\mathrm{PR}_i} + T_{\mathrm{TR}_i}) + \sum_{i=1}^{L_{\mathrm{A}}} (T_{\mathrm{AR}_i} + T_{\mathrm{A}_i} + T_{\mathrm{DR}}) + \\ + \sum_{j=1}^{L_{\mathrm{A}}} (T_{\mathrm{DR}} + T_{\mathrm{A}_j} + T_{\mathrm{PR}_j} + T_{\mathrm{TR}_j} + T_{\mathrm{AR}_j}) \quad (3)$$

where:

T_{NCT} – network cycle time,
L_{A} – number of devices,
T_{TR} – token transmission time,
T_{DR} – token detection time,
T_{AR} – time of analysis of token,
T_{PR} – time of preparation of token,
T_{PR_i} – time of preparation of data package,
T_{TR_i} – time of data package transmission,
T_{A_i} – cycle time (complete processing time for all actions in device),
T_{AR_i} – time of data package analysis,
T_{DR} – time of package detection,
T_{TP} – time of acknowledgement transmission.

Ethernet is an example of a protocol with random access. There are no constant delays and all stations transmit on demand. In this case network cycle time has to be calculated with probabilistic methods. In a normal situation, protocols are quick enough. In case of an alarm situation (cascade of events), these protocols cannot provide a reliable and efficient transmission of the data within, there are no mechanisms for controlling of exchanges. Efficiency of Ethernet protocol is dependent on network traffic, on small traffic is much better than in Master-Slave protocols but in heavy traffic is much worse [12].

4 Service Oriented Architecture

Today, SOA is one of the most successful paradigms in information systems. It gives many benefits like: openness, modularity, scalability and interoperability. These properties are also important in manufacturing systems. However, connecting SOA (which is based on Event-Driven Architecture) to the control systems requires additional software. There are many projects [1,2,3] with description of rules and methods on how to implement system for a complete automation. A complete industrial automation consists of planning, preparation, reporting, monitoring, control and regulation. For the ERP and MES, planning and reporting are the most important. Application on these levels should be universal and available on all software platforms for users at different locations. In SOCRADES [1], interfaces and procedures to create and configure control applications for distributed control system are defined. SOCRADES describes all levels of production process according to ISA 95 [16]: sensors and actuators, fieldbus, SCADA, local network, MES, ERP.

Specifications of services, interfaces, protocols on levels 2–4 (see Fig. 3) are presented precisely. For levels 0–1 there are only general guidelines such as: to use open protocol instead proprietary, if only this is possible or more versatile services needs more processing power. Due to lack of one, existing standard, (or due to too many standards) it is impossible to describe precisely integration of levels 0 and 1. Each protocol may have different way of connection. This is the reason that Service Oriented Architecture, which is proposed in this project, requires high computing power to provide proper connection.

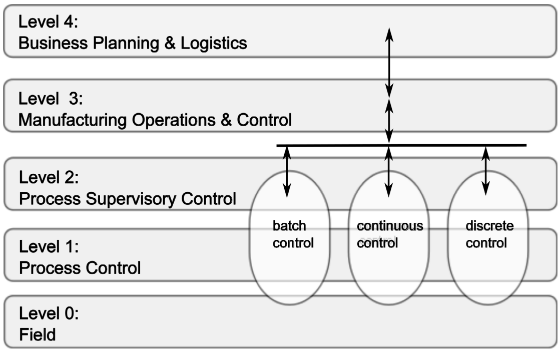

Fig. 3. ISO 95 model of production

Time is not the most important parameter of ERP or MES system. The situation is different for control processes, where time parameters (time delay and variations of time delay) are crucial. Time delay should be sufficiently short and stable. Reduction of time delay and more precisely reducing variations of time delay, increase the stability of the system and quality of control. However, increased time delay (variations of time delay) leads to loss of stability and to errors in control. For some protocols which require high speed of data transfer, using of SOA is not sufficient to achieve the desired integration goals. The authors of SOCRADES hope that, in near future, together with progresses in computer hardware, a total integration can be achieved.

The term fieldbus, used in the ISA 95 standard, can mean one of several protocols [17] on the lowest level of production. There are some methods for connecting the fieldbus level to the higher level. Because of big differences in characteristics and requirements of protocols used on these levels, there are no standard ways to connect (integrate) them yet.

One method is to use an agent – additional device connected physically to communication channels on both levels. An agent monitors all network traffic on the lower level and sends chosen, necessary data to the higher level. This method allows only a one-way communication. A different method is to use an interconnection device. It can be hardware or software device which allows to-ways communication (through conversion) between protocols. This method causes time delay from conversion and impact on industry network [6,7,18].

There are some methods for integration SOA system to the control system. There are two proposals: Devices Profile for Web Services (DPWS) and OPC Unified Architecture (OPC UA). DPWS enables UPnP which enables devices to use dynamic discovery, services description, messaging and events and subscriptions. There is only set of services for connecting and communication between separate devices, which include collection of data and security, but no standardization. OPC UA provides models for both whole system and services for proper operation. In addition OPC UA (as the only software package now) enables object oriented model. It means that not only data from manufacturing are collected and saved but relations and dependency between them too.

The differences are that DPWS was invented for direct communication between devices (levels 0 and 1 in ISO 95) [19] and OPC UA evolved to integrate different measurement and relations in a system (levels 2, 3 and 4) [20].

For further consideration OPC UA was chosen. It has the possibility to map full industrial processes with access through standard interfaces. It is possible to make independent objects with measurements and relations, services to discover, read, write, subscribe with secure channels. It can be used in interconnection of different standards such as ISO 15926 [21].

5 OPC UA

OPC UA has many implemented services. There are services between server and client such as browsing, subscription and read/write enables full control over process. A client application, without prior complicated preparation, can get list of variables, status and configuration, and even change values in process. The only condition is to have TCP/IP protocol. In theory, any client application, from any place, can assist in control.

Other services can be used in server communication to synchronize data, to improve safety by providing redundancy or use so called discovery server. The discovery server works as a special server which collects all information about other servers. This server contains information about all measurements and relations between them (ontology). It can simplify discovery services by obtaining all information at a single location (Fig. 4).

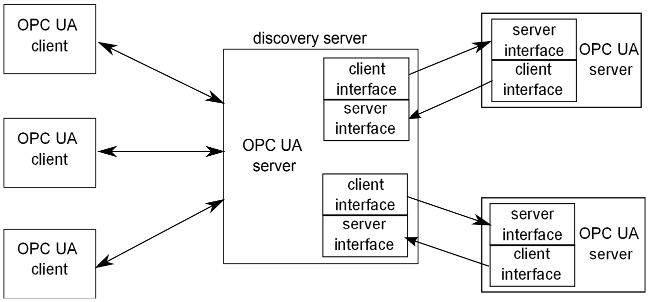

Fig. 4. Aggregating servers

A system consists of at least one OPC UA Server and many OPC UA Clients. All services and mapping are defined in standards.

6 Integration Problems

In theory, full integration of SOA and control systems is possible. In practice, the main problem is the impact of additional processing on control system time parameters. For several industry protocols, mathematical calculations can be used to find minimal network cycle time. There are no relevant equations for SOA systems. Integration has to be made with respect to:

– impact on network cycle time,
– sufficient processing power on OPC UA server.

Consider first problem. As was mentioned in Sect. 4, there are two types of integrations with fieldbus according to direction; one way, which can be made with agent device and two-ways, with specialized interconnecting device. The method with agent can be used in monitoring systems, where it is not necessary to change control system parameters. Interconnecting devices can be used in both types, but it adds additional delay. To calculate the time delay, Equation (2) and (3) can be used. Assuming that all devices acting in control are the same and processing, detection, transmission times are much shorter ($< 1\,\text{ms}$) than PLC processing time (typically 5–100 ms), it is possible to evaluate network cycle time.

The simplified network cycle time depends proportionally on number of devices and device processing time.

$$T_{\text{C}} = K * L_{\text{A}} * T_{\text{A}}$$

where:

K – constant value,
L_{A} – number of devices,
T_{A} – device processing time.

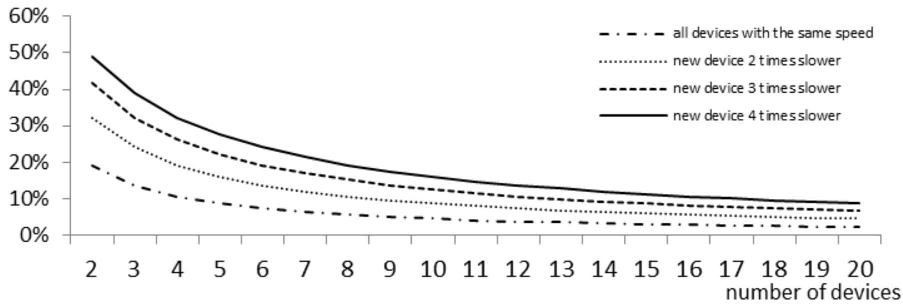

Fig. 5. The network cycle time extension after connection of the next device

The more devices are connected, the longer is cycle time, and typically PC has much longer processing time than PLC, which causes longer cycle time what is presented on Fig. 5.

The second problem concerns sufficient processing power on OPC UA Server. The server is connected to industry and to local network. In experiments [22] impact of both networks was measured. First, different methods for updating data in OPC UA server were tested. Updating data, without internal queue on server, causes heavy processor load and leads furthermore to server time parameter exceeding (Fig. 6).

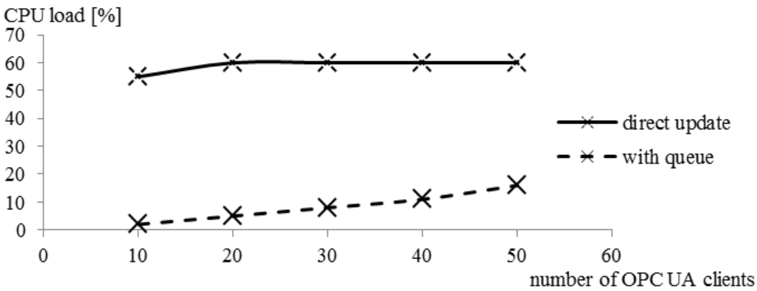

Fig. 6. Direct vs indirect updating of Servers data model [22]

In next experiment, dependency between server processor load and amount of the clients and amount of the subscribed data was measured (Fig. 7). Critical is amount of transferred data. Server has boundary for processed data amount, if data amount exceeds limitation, server cannot work properly. The boundary is dependent on hardware, operating system or application type. Solution to boundary problem can be dividing whole data processing from one OPC UA server into several OPC UA servers. The division causes reducing processor load on each server, but increases traffic load on industry network. Each of the servers has to obtain data from the control system, and thus increases network cycle time.

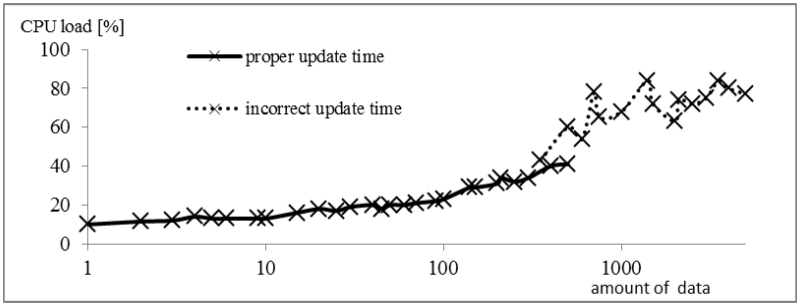

Fig. 7. Model of control system with network delays

7 Conclusions

New technologies like SOA, introduce new possibilities in information systems and maybe also in automation. Manual configuration of devices can be replaced with standard and open service sets. Due to object oriented model in OPC UA, it can be possible to get information not only about one measuring, but about all related measurements. The services should be independent of control system and hardware. It can reduce developing time and costs. But full automation is not always possible. There are two main problems: network cycle time and processing speed. Both problems depend on additional parameters such as type of protocols, amount of devices, time restrictions, etc. Today, there is no standard solution which can configure all parameters automatically.

In practice, complete integration is not always necessary. Authors suggest combination of two methods: use agent(s) to get all necessary data from control system, and one interconnection device with pre-defined parameters, which are allowed to change. This solution gives all possibilities for browsing and subscription of data and some (limited) possibilities to parametrize control system.

Acknowledgements. The authors thank Mr. Preben Gråberg Nes for his advice and help.

References

1. Taisch, M., Colombo, A.W., Karnouskos, S., Cannata, A.: SOCRADES Roadmap: The future of SOA-based factory Automation (2009)
2. Delsing, J., Eliasson, J., Kyusakov, R., Colombo, A.W., Jammes, F., Nessaether, J., Karnouskos, S., Diedrich, C.: A Migration Approach towards a SOA-based Next Generation Process Control and Monitoring. In: 37th Annual Conference of the IEEE Industrial Electronics Society (IECON 2011), Melbourne, Australia (2011)
3. SOA in Manufacturing. Guidebook, White Paper. A MESA International, IBM Corporation and Capgemini (May 2008)

4. Nilsson, J.: Real-Time Control Systems with Delays. Ph.D. Thesis, Department of Automatic Control, Lund Institute of Technology, Lund (1998)
5. Ogunnaike, B.A., Ray, W.H.: Process Dynamics, Modelling, and Control. Oxford University Press (1994)
6. Addad, B., Amari, S.: Delay Evaluation and Compensation in Ethernet-Networked Control Systems. In: 16th International Conference on Real-Time and Network Systems, pp. 139–148 (2008)
7. Lian, F.-L., Moyne, J., Tilbury, D.: Network design consideration for distributed control systems. IEEE Transactions on Control Systems Technology (2002)
8. Florescu, O., de Hoon, M., Voeten, J., Corporaal, H.: Probabilistic modelling and evaluation of soft real-time embedded systems. In: Vassiliadis, S., Wong, S., Hämäläinen, T.D. (eds.) SAMOS 2006. LNCS, vol. 4017, pp. 206–215. Springer, Heidelberg (2006)
9. DeVan, W., Hicks, S., Lawson, G., Wagner, W., Wantland, D., Williams, E.: Using a control system ethernet network as a field bus. In: Proceedings of 2005 Particle Accelerator Conference. Tennessee, Knoxville (2005)
10. Kwiecień, A., Sidzina, M.: The Method of Reducing the Cycle of Programmable Logic Controller (PLC) Vulnerable "to Avalanche of Events". In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2011. CCIS, vol. 160, pp. 379–385. Springer, Heidelberg (2011)
11. Hespanha, J.P., Naghshtabrizi, P., Yonggang, X.: A Survey of Recent Results in Networked Control Systems. Proceedings of the IEEE 95(1), 138–162 (2007)
12. Gaj, P., Jasperneite, J., Felser, M.: Computer Communication Within Industrial Distributed Environment – a Survey. IEEE Transactions on Industrial Informatics 9(1), 182–189 (2013)
13. CAN network – ISO 11898 standard
14. Communications Networks. Programming Manual. Cegelec, Clamart (1993)
15. Open MODBUS/TCP Specification. Schneider Electric (1999)
16. ISO 95 standard
17. IEC 61135 standard
18. Barbosa, R.R.R., Sadre, R., Pras, A.: Difficulties in Modeling SCADA Traffic: A Comparative Analysis. In: Taft, N., Ricciato, F. (eds.) PAM 2012. LNCS, vol. 7192, pp. 126–135. Springer, Heidelberg (2012)
19. OASIS Devices Profile for Web Services (DPWS) Version 1.1
20. OPC Unified Architecture standard
21. Sande, O., Fojcik, M., Cupek, R.: OPC UA Based Solutions for Integrated Operations. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2010. CCIS, vol. 79, pp. 76–83. Springer, Heidelberg (2010)
22. Fojcik, M., Folkert, K.: Introduction to OPC UA Performance. In: Kwiecień, A., Gaj, P., Stera, P. (eds.) CN 2012. CCIS, vol. 291, pp. 261–270. Springer, Heidelberg (2012)