INTELLIGENT VISUALISATION OF PROCESS STATE USING SERVICE ORIENTED ARCHITECHTURE

Jari Seppälä * **Mikko Salmenperä** *

* *Tampere University of Technology, Institute of Automation and Control, P.O. Box 692, FIN-33101 Tampere, Finland*

Abstract: Modern maintenance personel are becoming experts that are rare in number on global scale. The common factors with these experts are 1) they are on the move; 2) they are utilised in global scale; 3) they have multiple information access devices with different capabilities and 4) they rely on information on demand with extensive meta information requirements. The automation services support system is in the future integral part of core automation system and works as extension of the experts. It provides preprosessed information and solution suggestions for the mobile expert. Process state information provided to the maintenance personnel must be in compact form. As information in automation services support systems comes more and more globally distributed becomes Service Oriented Architechture viable solution for information access. Mobile experts require compact and fast decision supporting displays and user interface in order to handle the increasing work load. This requires intelligent, intuitive and robust preprosessing system as a backbone for automation lifecycle management. This paper presents how service oriented architecture can be utilised in automation services support system. The introduced framework combines rapid analysis development and intelligent process state visualisation for mobile expert and discusses the challenges met in building reliable mobile services for global automation.

Keywords: Visualisation, Fuzzy Logic, Temporal shapes, Web Services, Security, Automation Services Support System, Service Oriented Architecture, Mobile.

1. INTRODUCTION

Modern automation contains very complex systems. Fortunately and unfortunately they are extensively equipped with sensors and measurement networks. The complexity increases the number of sensing points in the machine. For example, in the industrial paper machine over 250 sensor channels generate a measurement every minute. The selected channels for normal process station display are only a small part of all collected measurement channels. Of course, more and more often this should be done every second. The task the operator faces in everyday life is how to find the essential information from the vast number of automation system and sensor displays.

The problem of increasing complexity results in decreasing number of experts that have the knowledge for process analysis and maintenance. These experts gradually become global in nature. Their expertice is required all over the world and their annual travel days is exploding. The effective usage of expert knowledge and time requires making the personnel mobile and making the information systems to support the mobile users.

The trend in automation services moves towards automation system lifecycle management. Automation system providers do not sell maintenance anymore rather they sell total lifecycle management. The main competitive factor for system providers is uptime and increase in system uptime requires fine tuned usage of resources and assets. The key resource is the system expert together with automation services support system.

Service Oriented Architecture (SOA) is not a new idea rather it is an old one. Reason for it's new coming can be found from Web Services technology, that is rapidly maturing into a de-facto standard in Internet enabled services. Web Services are also a natural way to integrate different systems together as they provide clear way to define interfaces between services, abstract data model for the exchanged data and isolation from underlying operating system and programming language.

The goal of this research was to create a framework utilising SOA for distributed and intelligent analysis with mobile maintenace personel friendly visualisation of process state. The prototype consists of a data acquisition service, a mathematical analysis service, a fuzzy logic engine for operator knowledge embedding, and a visualisation service. Analysis and fuzzy logic service are run on Matlab® and actual visualisation is done on the mobile device. System is constructed using SOA principle together with secure Web Services. These enable reliable distribution and loose coupling of different services in global scale.

2. PROPOSED CONCEPT

Current process monitoring and control systems are not well designed for predicting faults. The ability of building a intelligent fault prediction system into a process control system requires rather a process information system built on a more open database than current type of process monitoring system which includes only few extraction and interaction features of common databases. The links to external systems are also dependent on the system provider. The trend is moving towards open systems from the current closed ones. The term open in this context means usage of standardized and well documented interfaces like Java Data Base Connectivity (JDBC), OLE for Process Control (OPC) or Structured Query Language (SQL). The introduction of, for example, NELES Distributed Network of Applications (NelesDNA®) type of process control systems together with spreading use of OPC and

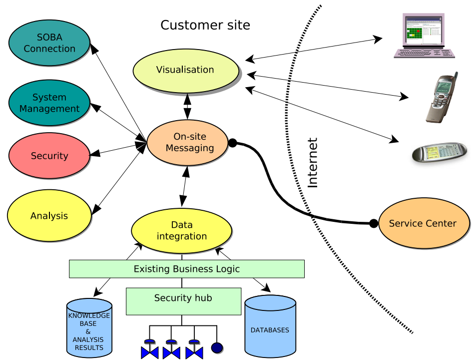

Fig. 1. Generic design of a monitoring network based on SOA

SQL enables easy development of additions to process information system.

Any service added to automation network has several key feature requirements which are connectability, message transfer reliability, security and loose connectivity. Connectability means interfaces to the various systems. This can be achieved by using messaging and messages within the system that are based on open standards such as XML and SOAP (Haavisto, 2001). The Second requirement is message transfer reliability. All message chains inside, as well as to and from, the system must be reliable and traceable. There are message types that must not be lost under any circumstances. The basis of messaging reliability can be achieved by utilising Message Oriented Middleware (MOM) (Kero, 2004) or products that offer similar functionality (Salmenperä, 2000). Security is not a feature requirement - it must be a integrated part of the system and is further studied in references (Nikunen, 2001) and (Salmenperä and Seppälä, 2004). The last but not least requirement is loose connectivity. In modern distributed automation environments customer sites and service providers must not have permanent connection dependencies. Also static binding of services must not occur i.e. you cannot use static IP addresses in any referencing to services as the address spaces tend to change every now and then.

Figure 1 depicts a rough design of how monitoring could be done. It shows how the services are loosely connected via MOM used as a backbone of all messaging. This kind of a common framework is the essential part of the future automation network (Nikunen and Koivisto, 2001; Salmenperä and Koivisto, 2003).

The concept presented in this paper conforms to Service Oriented Architecture programming paradigm. Each service is implemented as a web service and provides certain functionality to the system. When combined theses services produce

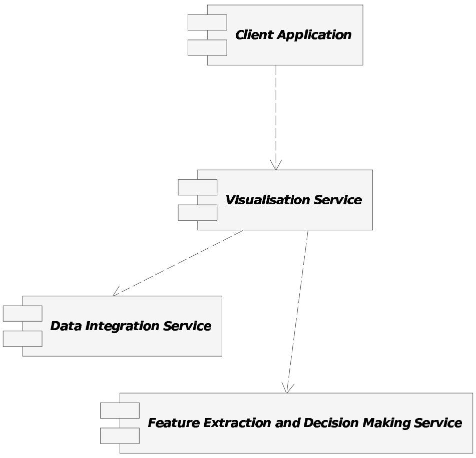

Fig. 2. SOA based architecture implies distribution and provides a natural way to develope services separately.

the final application. SOA and web services isolates the implementation and developement of different services. Only this visible from the service is the interface description in WSDL-format. For example, the analysis part with the data pre-processing and the feature extraction can be developed and run in complete separation from the rest of the system. In fact it could be outsourced from external service provider specialising in process analysis and optimisation. Book (Linthicum, 2004) shows general concepts behind SOA and Web Services in application integration.

The Visualisation service seen in the Figure 2 utilises two other services. This provide necessary modularity and flexibility into the application. The Visualisation service acts as coordinator between Data Integration and Feature Extraction and Data Analysis services. It also provides visualisation results for the user in a format suitable for current end device.

The main idea behind the analysis concept is to detect the measurement changes and to make fault prediction partly at the sensor level automatic thus enabling fast detection and reporting of the problems. The data analysis is done on Matlab® software which collects information from process valve (Helanterä, 2004; Helanterä *et al.*, 2004).

2.1 Data Integration Service

Data Integration Service is responsible for providing online/offline data for the system. It is in essence the gateway to underlying automation and information systems abstracting vendor specific data aquisition methods and providing general presentation of data.

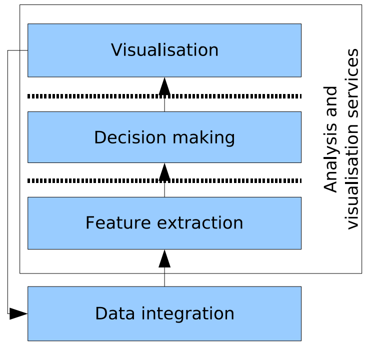

Fig. 3. The different services of the analysis and visualisation system. All the layers can be implemented as a separate service which can be run in a thread or computer depending on the target environment. The services utilise Web Service interface.

Public interface of the Data Integration service was chosen to conform to OPC XML DA-specification, which is Web Service based implementation of regular OPC DA. This way the integration of any existing OPC servers can be handled with a simple gateway program performing conversion between OPC DA and OPC XML DA. As OPC XML DA is a full web service it has a clear interface description which can be used to integrate other (non OPC) data sources into the system by simply implementing this interface into existing system or by creating a gateway program performing transtalion from proprietary format into OPC XML DA.

2.2 Analysis and Visualisation

The visualisation environment is designed to be modular (Seppälä *et al.*, 2000). The modularity enables the implementations of different layers in different parts of the automation system, Figure 3. For example, the analysis part with the data pre-processing and the feature extraction can be run in a separate workstation and even be implemented in the database, if the database supports this, to reduce the computing time. The decision layer can run in the mobile device within the visualisation application or as separate service in e.g. visualisation computer. In the last case only the information needed for visualisation is transfered to mobile device. This modularity enables also the separate development of each service as well as distributing the application throughout the Automation Service Support System network.

2.2.1. Feature extraction layer The mathematical analysis engine collects a simple analysis tools for online analysis, e.g. trend analysis

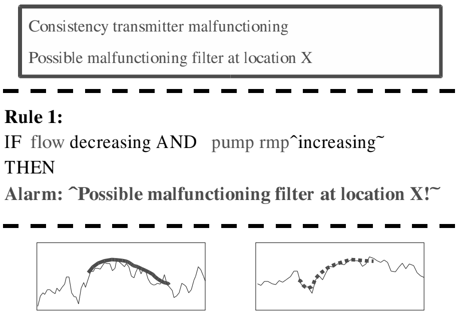

Fig. 4. Example analysis and visualisation process. Below a polynom is fitted on the data (feature extraction). On the middle layer Fuzzy logic is used for reasoning (decicion making), and finally the strongest candidate is presented to the user (visualisation).

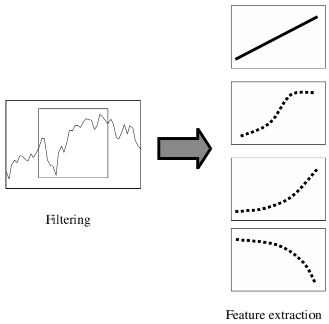

Fig. 5. The primitives are fitted into a selected data window. All features were presented as primitives such as 'increasing steadily' (line) or 'decreasing concavely' (a second order equation).

(Kivikunnas, 1998; Kivikunnas, 1999) and statistical process control analysis, and utilizes temporal shapes (Konstantinov and Yoshida, 1992; Rengaswamy and Venkatasubramanian, 1995) which are used to detect different process states and features (Paunonen, 1997). The selection of the data window is the most demanding part of the analysis layer. The window size depends on the signal and the target phenomena we are trying to find. In our case the window size for each channel was heuristically selected. The feature extraction layer can also be embedded into automation devices. In this case, the decision layer receives encoded features from devices thus decreasing the computational requirement of decision making process and increasing decision making speed.

2.2.2. Decision layer The operator knowledge embedded into the decision layer can be collected via interviews and discussion sessions where the developed rules are analyzed and commented by

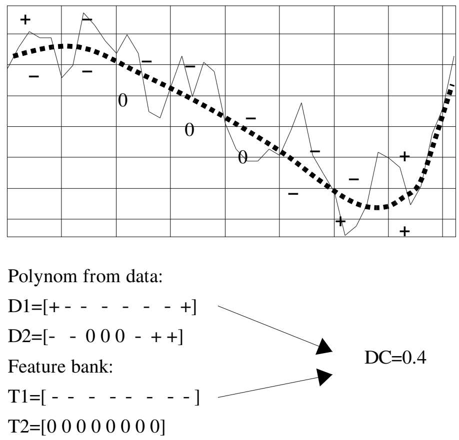

Fig. 6. The best describing primitive is selected by fitting a polynom on the data window. Then the First and Second derivative are calculated from the polynomial and these are then compare with the primitives in the feature bank.

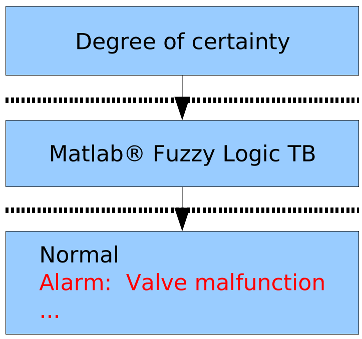

Fig. 7. The degree of certainty is fed into the rules implemented in Matlab Fuzzy Toolbox.

the operators (Paunonen, 1997). The decision-making, or the 'intelligence' in the feature combining phase, is implemented using fuzzy logic. In uncertain cases where operator knowledge is not available linguistics equations (Juuso, 1999) are feasible solution. The fuzzy logic engine is essential to be able to find the unusual behavior from the normal phenomena induced by the process. The rules are based only on the degree of certainty value. This proved to be inadequate for making efficient rules for paper machines (Seppälä *et al.*, 2000). However, for presented hardware monitoring it is enough.

2.2.3. Visualisation layer The mobile visualisation interface must be kept light and yet it must provide the user enough information for rapid

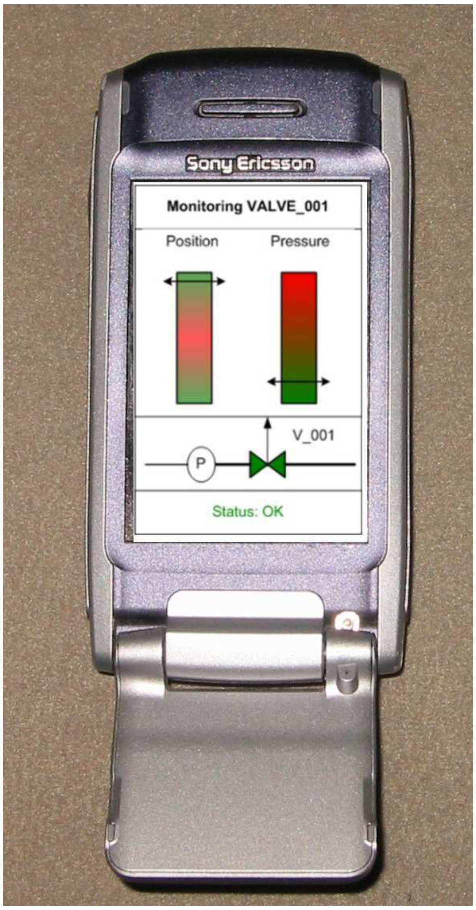

Fig. 8. Example of the final visualisation environment for the end user.

decision making. In this case, the visualisation consists only simple model of the process valve. The histograms give overall process information "at a glance" and are displayed beside the valve. In full scale visualisation environment the valve in Figure 8 is brought into experts attention by automatically navigating in process hierarchy.

2.3 The development tool

The development tool is designed for a research scientist who will implement the rules into the system. The Figure 9 above illustrates a highly simplified presentation of the tool and the details are left out to clarify the idea. The tool consists of 1) data window into which the current data as well as the fitted models are plotted; 2) the eight small squares present the first and second derivatives of the fitted model; 3) the alarm window will show the activated alarm and 4) the eight small buttons on the right hand side are used for starting the simulation, editing rules and data handling.

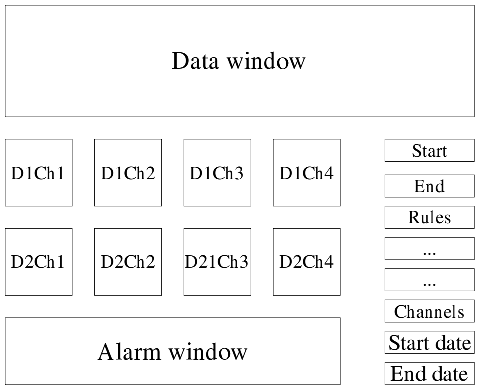

Fig. 9. The simplified layout presentation of the developers tool.

3. CONCLUSIONS

An intelligent visualisation framework using Service Oriented Architechture was developed. The analysis and visualisation system consists of four main layers which can be easily distributed over the network; 1) the data acquisition; 2) the mathematical analysis and feature extraction layer; 3) the decision making layer and 4) the visualisation layer. These layers are interconnected using secure Web Services which enables the distribution of all layers to different parts of the Automation Services Support System. The analysis, feature extraction and decision making is built using commonly used mathematical software which simplifies system development process. The visualisation on the mobile devices is implemented keeping in mind the portability requirements and uncertain networks.

This type of fault detection, prediction and reasoning system will be the future trend in automation industry. There are still many problems to be solved in the information and measurement gathering systems to enable the efficient usage and development of this type of analysis and visualisation environment. However, the demand of mobile expert support features instead of plain fault reporting systems is growing with the increased of complexity automation environments and multinational nature of automation service providers.

The introduced framework supports global condition monitoring services and is designed to ease the work of the decreasing number of system experts on the move. The implementation of the presented visualisation system requires good commitment on behalf of the automation process personnel and/or expert organisations. The knowledge needed for building the reasoning layer cannot be done in any other way than in close contact with the automation experts. The ideas presented will be implemented in the future Automation Services Support Systems. The future improvements include building semantic framework for gathering the silent information from the experts and designing light model for reprecenting automation systems in the mobile devices.

REFERENCES

Haavisto, Henri (2001). .NET messaging techniques in condition monitoring network. Master's thesis. Tampere University of Techonology, Automation Department. In Finnish.

Helanterä, H. (2004). Analysis services in global condition monitorin system. Master's thesis. Tampere University of Technology, Automation Department. Confidential, In Finnish.

Helanterä, H., M. Salmenperä and H. Koivisto (2004). Global condition monitoring system - implementing MATLAB-based analysis services. *First international conference on informatics in control, automation and robotics (ICINCO2004)*.

Juuso, E. (1999). Modelling and control using linguistic equations. *European Symposium on Intelligent Techniques (ESIT'99)*. http://www.erudit.de/erudit/events/esit99/12536_p.pdf.

Kero, J. (2004). Advanced messaging in enterprise scale maintenance system. Master's thesis. Tampere University of Technology, Automation Department.

Kivikunnas, S. (1998). Integration of process trend analysis with interval algebra. *EUFIT98*.

Kivikunnas, S. (1999). Overview of process trend analysis methods and applications. *Proceedings of Workshop on Applications in Chemical and Biochemical Industry*.

Konstantinov, K. B. and T. Yoshida (1992). Realtime qualitative analysis of the temporal shapes of bio process variables. *AIChE Journal* **28**(11), 1703–1715.

Linthicum, David S. (2004). *Next Generation Application Integration - From Simple Information to Web Services*. Addison-Wesley.

Nikunen, H. Salmenperä, M. and H. Koivisto (2001). Global condition monitoring network. *Automation Days 2001*.

Nikunen, J. (2001). Security considerations on wide area networking industrial solutions. Master's thesis. Tampere University of Technology, Automation Department.

Paunonen, H. (1997). Roles of Informating Process Control Systems. PhD thesis. Automation and Control Institute, Tampere University of Technology.

Rengaswamy, R. and V. Venkatasubramanian (1995). A syntactic pattern-recognition approach for process monitoring and fault diagnosis. *Engng. Applic. Artif. Intell.* **8**(1), 35–51.

Salmenperä, M. (2000). E-speak in enterprise scale condition monitoring network. Master's thesis. Tampere University of Technology, Automation Department.

Salmenperä, M. and J. Seppälä (2004). Data security considerations in modern automation networks. *First international conference on informatics in control, automation and robotics (ICINCO2004)*.

Salmenperä, M. Seppälä, J. and H. Koivisto (2003). Applying .NET framework to conditioning monitoring in globally distributed environment. *Automation Days 2003*.

Seppälä, J., A. Nissinen, H. N. Koivo and M. Laitila (2000). Intelligent visualisation of dynamic process data. *Control Systems 2000*.