Dynamic Alarm Management in Next Generation Process Control Systems

Eva Jerhotova, Marek Sikora, Petr Stluka

Dynamic Alarm Management in Next Generation Process Control Systems

Eva Jerhotova, Marek Sikora, and Petr Stluka

Honeywell Prague Laboratory
V Parku 18, 148 00 Prague, Czech Republic

{Eva.Jerhotova, Marek.Sikora, Petr.Stluka}@Honeywell.com

Keywords: next generation DCS/SCADA, system of systems, process monitoring and control, alarm management, dynamic alarming, state-based alarming, alarm load shedding, software-oriented architecture, SOA, complex event processing, CEP, IMC-AESOP project

Abstract. Current process control systems are composed of a large number of components and subsystems operating at different layers of the control system architecture model (i.e. measurement and control devices, Distributed Control Systems, Advanced Process Control systems, and Manufacturing Execution Systems). The IMC-AESOP project aims at designing the next generation architecture of process automation systems. In order to ensure system scalability and modularity, the new architectural design follows the SOA (Service-Oriented Architecture) design principles. Moreover, the design assumes adoption of various technologies with the aim to enable the control systems to meet all functional and performance requirements. The CEP (Complex Event Processing) technology has been selected for being able to provide efficient asynchronous communication (within and across architecture layers) and the capability of temporal reasoning over large amounts of system-generated events. This paper describes the intermediate results of the IMC-AESOP project, outlining the architectural concepts related to the use of the SOA and CEP technologies in the context of advanced alarm management applications - alarm load shedding and state-based alarming.

1 Introduction

Current industrial process control and monitoring systems – DCS/SCADA – are becoming increasingly complex and heterogeneous [1]. Communication among their components is evolving from synchronous to asynchronous. In the system architecture design, multiple key challenges need to be appropriately addressed, such as interoperability, real-time performance, security or availability. The European R&D-project IMC-AESOP (ArchitecturE for Service-Oriented Process - Monitoring and Control, http://www.imc-aesop.eu) proposes consistent use of Service-Oriented Architecture (SOA). This architectural approach assumes that all autonomous and intelligent de-

vices, applications and systems that are deployed nowadays or in near future will expose their capabilities, functions and structural characteristics as services. Under such paradigm, the industrial process environment will be mapped into a "Cloud of Services" where devices and applications distributed across different layers of the enterprise-control system hierarchy expose the services that they provide and at the same time they are able to access and use other services located in the cloud.

The SOA approach [12] has been widely adopted by enterprise architects for a number of reasons, which include increased flexibility in application design, providing opportunity of functionality reuse at the macro level (or the service level) rather than at the micro level (or the class level), and making the relevant businesses more agile and capable to respond more quickly to changing market conditions. The IMC-AESOP project aims at bringing the SOA architectural approach into the industrial automation world. A considerable part of the project efforts has been invested into describing the service-based architecture of a DCS/SCADA system and identification of services which could be exploited in multiple environments or applications. Some of the services are rather specialized, such as configuration services, data acquisition services, or alarm processing services, while other services may be more generic, such as discovery (e.g. service registry) or event broker services.

In the recent years, Complex Event Processing (CEP) [2] has gained considerable importance as a means to extract information from distributed event-based (or message-based) systems. It became popular in the domain of business process management but is now being applied in the industrial monitoring and control domains. This method enables efficient asynchronous communication (both within and across architecture layers) and a temporal reasoning functionality capable of processing large amounts of system-generated events in real time. CEP allows control systems to evolve from scan-based systems to event-based systems, which significantly increases the system performance and data throughput. In the IMC-AESOP project, Microsoft StreamInsight is used as a CEP engine. This engine is wrapped in an alarm processing service, which is being reused in different applications or use cases.

The remainder of the paper is organized as follows. Section 2 describes the advanced alarm handling techniques, whose specific implementations using SOA and CEP are being explored in the project. Section 3 summarizes the underlying architectural principles. And finally, section 4 discusses implementation challenges and plans for validation.

2 Advanced Alarm Handling Techniques

Alarms used in industrial manufacturing facilities are sound and/or visual announcements of abnormal process situations addressed to operators. These announcements are typically triggered when a given process variable (measured by a dedicated sensor) exceeds a predefined limit. Alarms require operator assessment and action in order to avoid (or at least mitigate) plant upsets, which may otherwise cause injuries, loss of production or equipment damage [6, 7, 8].

The alarming capability is ensured by the alarm system within the plant control system. As a consequence of process or equipment changes over time (e.g. equipment addition or replacement) the alarm system requires maintenance to sustain its best performance and full functionality. Alarm rationalization and continuous alarm system improvement [6,7,8,11] are the processes for achieving the goal of keeping the alarm system performance at an appropriate performance level during its life-cycle. During alarm system maintenance, certain alarms may be configured or de-configured and alarm parameters, such as limits, priority, deadband, or delay timer, may be changed.

Despite considerable investments into alarm system improvement efforts which have been made by operating companies, the performance of the alarm system in plant upsets remains a challenge and operators still experience alarm floods[1]. Occurrence of alarm floods can be mitigated by the use of advanced alarm handling techniques [6, 7, 8, 11], such as

- **alarm grouping** (i.e. presenting related alarms as a group),
- **state-based alarming** (i.e. using different alarm settings or suppressing alarms for different process or equipment states). This technique is sometimes called mode-based alarming, dynamic alarm masking, alarm suppression, or automatic alarm shelving
- or **alarm load shedding** (i.e. displaying only the most critical alarms during alarm floods and delaying presentation of alarms of less critical).

The phenomenon of alarm flooding is one of the pain points in the process industry and it was one of the main driving forces for establishing dedicated professional consortia, such as the Abnormal Situation Management Consortium (ASMC). Within the IMC-AESOP project, the capabilities of SOA and CEP are demonstrated on the state-based alarming and alarm load shedding methods that have a great potential for alarm flood reduction.

2.1 State-Based Alarming

In certain process states, static alarms can be inadvertently triggered due to normal process changes (e.g. different operating mode or equipment shutdown). In such situations, certain alarms become meaningless or their limits must be set too wide to accommodate the different states. State-based alarming is a dynamic alarm handling method based on switching the alarm system configuration to the settings which correspond to the identified process states. For the different states, new alarms may be enabled, certain alarms may be disabled or their parameters may be altered (such as the alarm priority or the alarm limit).

[1] An alarm flood is a situation in which the operator is presented with too many alarms than he/she can effectively respond to [7]. Such situations are potentially dangerous, since the operator may overlook important alarms or assess the situation wrongly because of stress and information overload.

The process states may be grouped into the following categories:

- **Unit or equipment shutdown**, such as stopping or shutdown of a redundant piece of equipment (e.g. when a redundant pump is turned off, all pump low-flow alarms are disabled), shutdown of a unit (e.g. for maintenance), or equipment out-of-service. Fig. 1 displays an example of alarm settings for shutdown of a tank. For the shutdown state, the low-energy alarms (e.g. low level in the tank) should be de-configured, while the high-energy alarms (e.g. high pressure) should not be de-configured, but only reconfigured to much lower values in order to be able to detect isolation failures and unexpected reactions from contamination [11].
- **Different operation modes**, such as start-up, steady operation, different feed, de-coking (e.g. higher temperatures for heater decoking).
- **Major event suppression** [8], such as suppression of a suction pressure alarm when the compressor trips off.

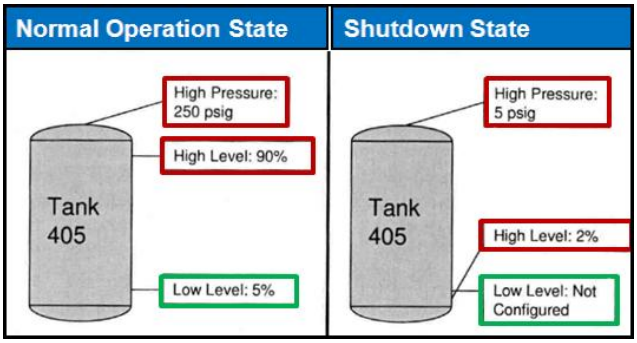

Fig. 1. State-based alarming for a shutdown state [11]

State-based alarming is becoming increasingly used in practice. This is true mainly for the shutdown-state alarming, for which the setup is relatively easy and many implementations already exist. In contrast, the other two groups of techniques are not currently much used because of requiring deeper analysis, such as the cause and effect analysis for consequential alarm suppression.

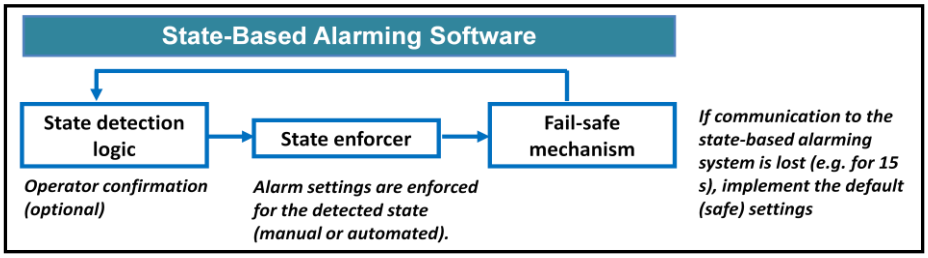

Fig. 2. State-based alarming software [11]

Regarding state-based alarming implementation, switching between states may be done manually (by the operator), semi-automatically (when the operator either identi-

fies or confirms the automatically identified state and initiates the configuration change), or automatically (when no input is required from the operator). For the automatic approaches, the state detection logic must be reliable and must not chatter. As depicted in **Error! Reference source not found.**, the state-based alarming implementation must include reliable alarm settings enforcement and a fail-safe mechanism. Under any circumstances, the operator must be kept informed about which alarms have been disabled or reconfigured and why and also should be able to override the state change.

2.2 Alarm Load Shedding

Alarm load shedding is a technique which supports operators in prioritizing actions in alarm flooding situations by displaying the most urgent alarms, postponing displaying of less important ones, and filtering out alarms of log priority and alerts. The aim of this method is to keep the alarm rate at a manageable level (ideally one alarm per minute [7]) as applicable.

There are two options for triggering this method: manual (by the operator who may select a preconfigured filter) or automatic (based on alarm flood detection). The former approach already occurs in the current practice, while the latter is not yet used.

All alarms should be available at operator's request (e.g. in the graphical display) and no delay must never be applied to emergency and critical alarms for safety reasons [7]. To yield good results, this method requires a rationalized alarm system with appropriate alarm limits and priority ranking, elimination of chattering alarms, eclipsing, or advanced alarm handling techniques, such as state-based alarming [6,7,8,9,11].

3 Advanced Alarm Management Using SOA and CEP

Alarm management functionality of a DCS/SCADA system is usually distributed among several system components including:

- **Automation devices** (or controllers) generating alarm events via systematic comparison of the actual values of all monitored points with the pre-configured alarm trip points (limits) and an alarm event is activated whenever specific trip point is reached
- A **DCS server** (connected to the supervisory control network) providing capability for automated logging (journaling) of alarm events.
- A **DCS station** (connected to the same network) through which the process operator interacts with the control system. The DCS station provides graphical displays visualizing a chronological list of active alarms (i.e. alarm summary), process schematics, and other types of displays supporting operator situation awareness.
- An **application node** representing a computer dedicated to alarm configuration, analysis and maintenance applications, which are usually designed for off-line use during alarm system improvement activities

The advanced alarm management techniques (described in section 2) are usually residing in the supervisory control network with individual functions shared between the server and stations. Several functional blocks (alarm configuration, alarm processing, event detection and others) are needed for successful implementation. Within the IMC-AESOP project, the advanced alarm management functions are realized through specific services or their combinations, as illustrated in Fig. 3.

- The **Sensory Data Acquisition** service delivers data from the process exploiting either the scan-based method (sending values regularly) or the event-based method (delivering values only when there has been any change)
- The **System Configuration** service provides information of the current process state (such as startup, normal operation, shutdown, maintenance, fault, off)
- The **Alarm Configuration** service lists all available alarms with their description, properties, and relation to process units and equipment

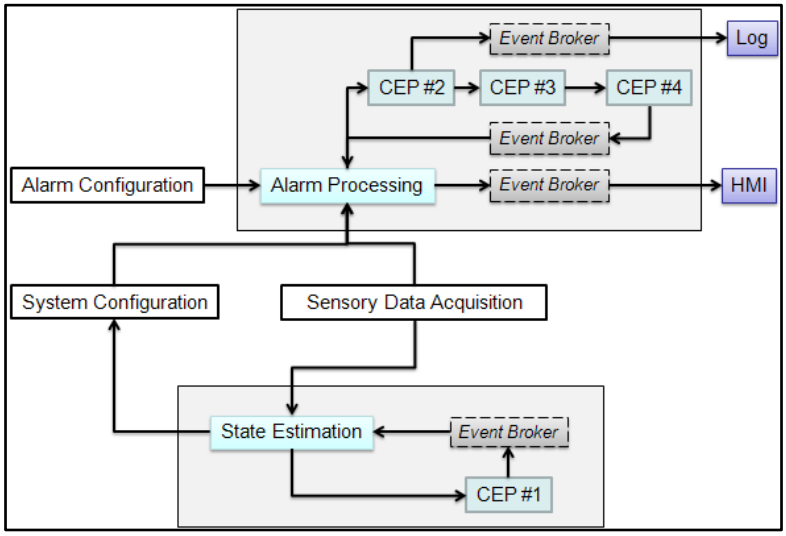

Fig. 3. Alarm processing architecture

There are also several instances of the CEP service that are being directly used by the alarm processing services. Each of them has the same StreamInsight service running in the background. The difference lies in the input and/or output events and the definition of the query (as described below). The following two services are the core of the SOA/CEP-based alarm processing application:

- Referring to Fig. 3, the **State Estimation** service uses the CEP#1 service, which evaluates predefined rules on the values of relevant process variables. For example, when a certain set of variables turns zero, it may be assumed that the associated unit has been turned off, or variables oscillation outside the predefined bounds may indicate that a corresponding unit is in a startup mode. The information about the identified state change of the unit is then stored.

- The **Alarm Processing** service represents an implementation of the alarm handling techniques described above. The CEP#2 service generates alarm events via comparing given process variables with their limits, while taking into account the process state identified by the State Estimation service. The CEP #3 service reduces alarm chatter by concatenating multiple closely-spaced alarms into a single alarm event. Finally, the CEP#4 service provides the alarm load shedding functionality. As illustrated in Fig. 3, the alarms identified by the CEP#2 block can be stored in the alarm log for possible further investigation, while the operator is presented only with the alarms output by the CEP#4 block. The SOA nature of this application also allows dynamic combining of the individual CEP alarm processing blocks. For instance, when only the state-based alarming functionality is implemented, the alarm load shedding block can be omitted.

Additionally, the **Event Broker** service can be used in the application architecture, which means that instead of sending the output events (i.e. alarms or state changes) directly to the consumer of the service, they are sent to the Event Broker service, from which any subscriber can be notified about the arrived events. In this architecture, the services may be more loosely-coupled and there may be multiple event recipients (e.g. the state change can not only be stored by the system configuration service, but can be also propagated to the operator to check if this change is correctly in place).

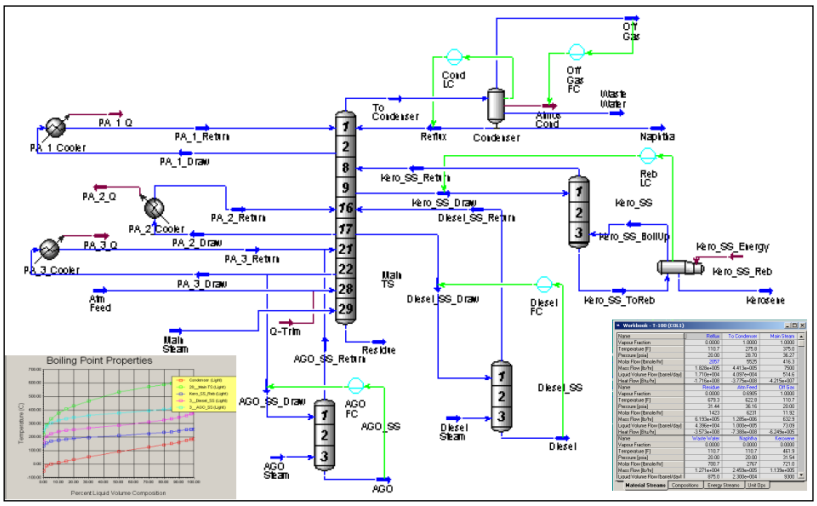

Fig. 4. Dynamic Unisim model of an atmospheric distillation column with three side strippers

4 Future Work and Conclusion

As the next step in the project, state-based alarming and alarm load shedding will be implemented to the process data generated by a Unisim model of an atmospheric distillation column [8] shown in Fig. 4. This facility produces naphtha, kerosene, diesel, atmospheric gas oil, and atmospheric residue products from a heavy crude feed. The

model is dynamic – i.e. comprising controllers and process specifications, such as the size of the distillation column trays and the side stripper tray sections, and the pressure flow. The incorporated dynamics allow for more realistic modelling of the facility behavior. Using this model, the process data will be generated by making changes to key process variables and also by simulating disturbances (e.g. failure of the reflux pump). The considered states for state-based alarming are shutdown, start-up, and different feed rate.

Based on the demonstration, we shall be able to assess the advantages and disadvantages of the SOA/CEP-based implementation for this type of applications. The main advantage of SOA lies in the loose-coupling of services which enables dynamic composition and orchestration. For CEP, it is the event-based functionality which yields improved performance and scalability. However, there are still a few open questions that need to be answered, such as how flexible the used CEP services really are, or if there will be any performance loss due to SOA or the Event Broker.

Acknowledgement

The research efforts described in this paper are co-funded by the European Union within the IMC-AESOP project (www.imc-aesop.eu). The authors would like to express thanks to the consortium partners for the fruitful discussions and cooperation.

References

1. S. Karnouskos and A. W. Colombo: "Architecting the next generation of service-based SCADA/DCS system of systems". Proc. 37th IEEE Conf. IECON, Melbourne, Nov 2011.
2. S. Karnouskos et al.: "A SOA-based architecture for empowering future collaborative cloud-based industrial automation", 38th IEEE Conf. IECON, Montreal, Oct 2012.
3. F. Jammes et al.: "Technologies for SOA-based Distributed Large Scale Process Monitoring and Control Systems" , Proc. 38th Conf. of the IEEE Industrial Electronics Society (IECON 2012), Montreal, Oct 2012.
4. A. Duca, N. Freeman, and S. Drews, "SOA What? Demystifying SOA for the Process Industry," Honeywell International Inc, Tech. Rep., 2008.
5. O. Etzion, P. Niblett, "Event Processing in Action", Manning Publications, USA, 2011.
6. D. Rothenberg, "Alarm Management for Process Control, A Best-Practice Guide for Design, Implementation, and Use of Industrial Alarm Systems," Momentum Press, 2009.
7. EEMUA, "Alarm Systems, A Guide to Design, Management, and Procurement," EEMUA Publication No. 191, London, 1999.
8. J. Errington, D. V. Reising, C. Burns, "Effective Alarm Management Practices," ASM Consortium Guidelines, ASM, May 2007.
9. P. Andow, J. Cade, R. Clark, W. Foslien, "Alarm Flood Analysis Report", Honeywell HPS, ASM Consortium, 2007.
10. J. G. Speight, "The Chemistry and Technology of Petroleum", 3rd edition, Marcel Dekker, Inc., USA, 1999.
11. B. R. Hollifield, E. Habibi, "Alarm Management: Seven Effective Methods for Optimum Performance", ISA, 2008.
12. T. Erl, "Principles of service design", Prentice Hall, 2007.