Chapter 9
Plant Energy Management

Stamatis Karnouskos, Vladimir Havlena, Eva Jerhotova, Petr Kodet, Marek Sikora, Petr Stluka, Pavel Trnka, Marcel Tilly

Abstract In the IMC-AESOP project, a plant energy management use case was developed to highlight advantages of service orientation, event-driven processing and information models for increased performance, easier configuration, dynamic synchronisation and long-term maintenance of complicated multi-layer solutions, which are deployed nowadays in the continuous process plants. From the application perspective, three scenarios were implemented including advanced control and real-time optimisation of an industrial utility plant, enterprise energy management enabling interactions with the external electricity market, and advanced alarm management utilizing the Complex Event Processing technology.

9.1 Introduction

Industrial operating companies have to pay increasing attention to monitoring and optimisation of energy efficiency and carbon emissions. In oil refineries and other enterprises in the petrochemical, chemical, pharmaceutical, or paper-making industry, the utility plant is responsible for the major supply of energy – primarily steam and power – to the process plant (as depicted in Fig. 9.1). The energy can either be generated locally, or purchased from an electricity distribution company. The industrial utility plants may have a contract allowing them to sell excessive amounts of energy back to the electricity grid and take advantage of variable tariffs. Depend-

Stamatis Karnouskos
SAP, Germany e-mail: stamatis.karnouskos@sap.com

Vladimir Havlena, Eva Jerhotova, Petr Kodet, Marek Sikora, Petr Stluka, Pavel Trnka
Honeywell, Czech Republic e-mail: vladimir.havlena@honeywell.com,eva.jerhotova@honeywell.com,petr.kodet@honeywell.com,marek.sikora@honeywell.com,petr.stluka@honeywell.com,pavel.trnka@honeywell.com

Marcel Tilly
Microsoft, Germany e-mail: marcel.tilly@microsoft.com

ing on local conditions, they can also serve as a source of heating for residential districts.

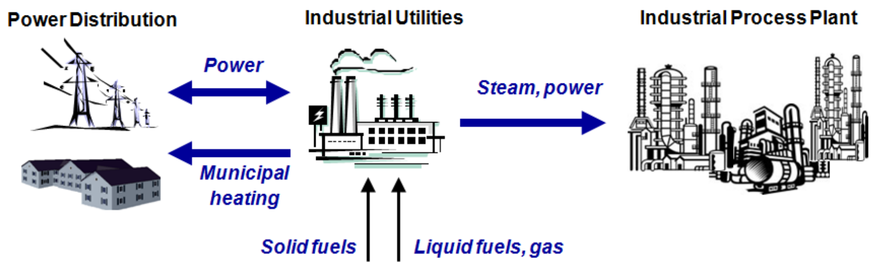

Fig. 9.1 Major flows of energy in industrial utility and process plants

Although both industrial utility and process plants are tightly interconnected, their operational and business objectives are different. In the utility plants, the generation of energy is the primary business objective, which is also consistently addressed throughout the facility by adopting hierarchical solutions for closed-loop control and real-time optimisation of individual pieces of equipment (boilers, turbines), their groups (several steam boilers connected to the same header), or the plant as a whole. In contrast to that, the process plants are primarily driven by the objective to produce appropriate mix of products to meet orders coming from the downstream industries.

Energy consumed in the plant and the cost of raw materials are the largest operating costs – and the general desire is to reduce this cost as much as possible, but never in a way that could threaten timely delivery of products. Coupling the industrial utility plants with the envisioned Smart Grid [13] infrastructure in real-time may enable new business opportunities for both sides and enhance energy efficiency. Industrial plants, which have the flexibility to adjust [7] production processes and/or objectives in response to the signals coming from the electricity market, could be seen as an integral part of a larger smart grid ecosystem.

The main objective of the work conducted in the IMC-AESOP project was to highlight advantages of service orientation, event-driven processing and information models for easier configuration, dynamic cross-layer synchronisation and maintenance of complicated multi-layer solutions, which are deployed nowadays in the continuous process plants. From the application perspective, the following three scenarios were implemented:

- energy management of an industrial utility plant enhanced through **cross-layer consistency management**, based on information models
- **adaptive enterprise energy management** enabling interactions with the external electricity market
- enhanced operation of processing units through a more effective **alarm management**, driven by the Complex Event Processing technology

9.2 Cross-Layer Consistency Management

9.2.1 Problem Description

The distributed and networked control of large-scale systems is typically de-signed as multi-layer control architecture, as is the case of industrial utility plants, whose operation is managed by implementing the following application layers:

- *Equipment level*: This basic level is focused on real-time optimisation of individual pieces of equipment – basically the pressure control related devices like boilers, let down valves and vents, but also other types of more complex equipment including turbo generators or condensing turbines. Advanced process control for boiler modulates fuel feed and air flow to boiler in order to maximize boiler efficiency.
- *Unit level*: Applications at this level deal with the problem of optimal allocation of load between several pieces of equipment running in parallel. This task is usually executed in real-time to ensure fast response to dynamically changing conditions and external requirements. Total steam production is allocated to individual boilers with respect to their efficiency curves to minimize the cost of steam production. The same approach may be applied to multiple turbines.
- *Plant level*: Applications optimize operation of the utility plant over significantly longer periods of time – ranging from hours to days – taking into account multiple possible configurations of the utility plant that can be selected for meeting the energy demand requirements.

The hierarchical approach brings the advantage of a simplified design for complex control strategies, but on the other hand, it complicates the information consistency between individual layers (PID controllers, advanced process controllers, real time optimizers) under changes in plant topology and other events. Each control layer requires different representation of knowledge, which makes it difficult to guarantee the cross-layer integration, consistency and uniform representation of on-line and off-line process data, topology information and performance models. Within the IMC-AESOP project, these challenges were addressed by implementing a two-level server architecture and OPC-UA [8] information model, which brought the following benefits:

- Information model consistency on all hierarchy levels
- cross-layer integration
- Event-driven consistent reconfiguration of all layers
- Support for flexible on-demand optimisation and what-if analysis
- Data aggregation from heterogeneous sources

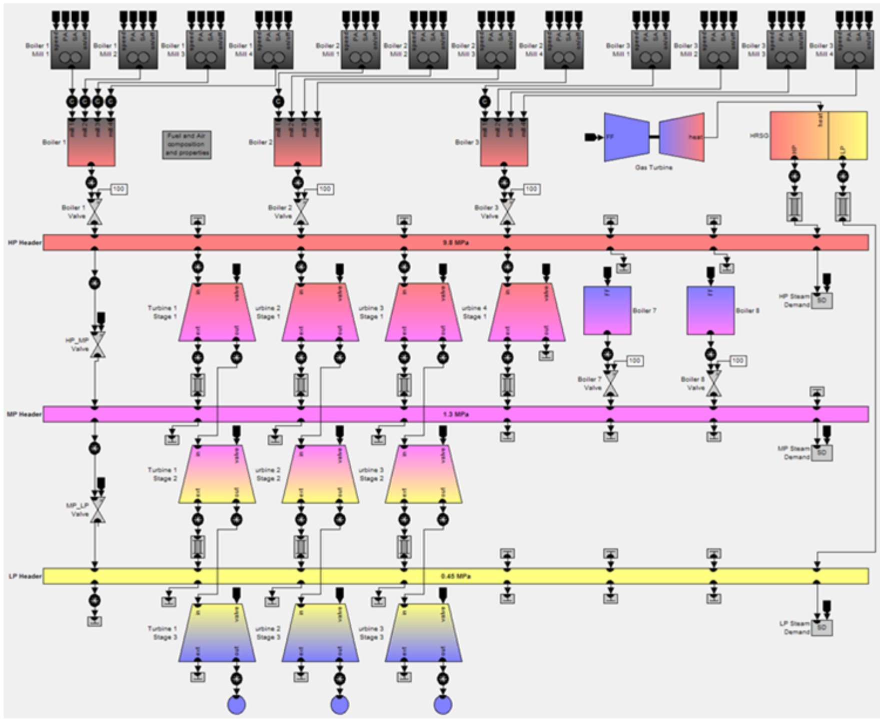

Fig. 9.2 Utility plant model in Matlab Simulink

9.2.2 Information Model

The presented application scenario was primarily focused on the design of information model for a large scale control system, which was demonstrated on a model of real industrial utility plant (Fig. 9.2). The two-level architecture integrated L1 and L2 information servers (Fig. 9.3) with complementary functions. Raw data collected from the utility plant were aggregated and unified by L1 servers, which are bound and act as a single virtual server containing full OPC-UA information model with data, metadata and topology information. The aggregating L1 server provides unified access point and event generation to chained L2 servers, which are specialized interfaces mapping L1 information model to a cloud of shared L2 services. The services are higher layers of control hierarchy: advanced process control, real time optimisation, scheduling, and business planning.

The information model was designed to consistently and uniformly represent on-line and off-line process information, which can be used by multiple users with different requirements and functionality. The users can be controllers on a different level of control hierarchy (PID controllers, advanced process controllers, real time optimizers), process operators using their operator screens, alarm management sys-

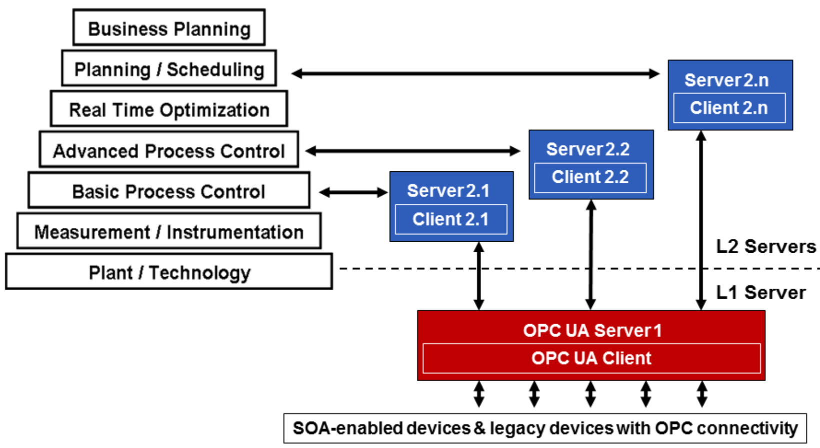

Fig. 9.3 Control hierarchy with Layer 1 and Layer 2 information servers

tems, process historian, etc. The information model has two consistent levels. The levels differ in their information details according to their target use.

L1 model is a low level model containing detailed process information and topology information, which is used by L1 server. The process is represented as a set of *devices*, which have *input* and *output* ports and these ports are interconnected by *streams*. The devices, ports and streams create basic framework of L1 model [9].

The additional details are usually directly associated with the objects of this basic framework. OPC-UA information model topology is not limited to a tree. It allows "full mesh" topology; however, the backbone of this full mesh structure is usually a tree of hierarchical references, which is used for nodes referencing. The topology information held by L1 model describes interconnections between devices, which always has the following pattern: $Device \rightarrow Port \rightarrow Stream \rightarrow Port \rightarrow Device$.

Specific object instances can also represent a process value measured by a sensor or a set point for an actuator. This object can be attached directly to a device (measurements or actuators folder) or to a port or stream as illustrated in Fig. 9.4.

L2 model is a high level model for advanced process control and other higher control layers (APC, RTO, and MES). It contains information necessary for retrieving dynamic or static behaviour models needed by higher control layers. It has dynamic and / or static model of individual devices (examples of model are state space linear model, transfer function matrix, non-linear static model, etc.). and includes description of ports role in controller design (MV – Manipulated Variable, DV –

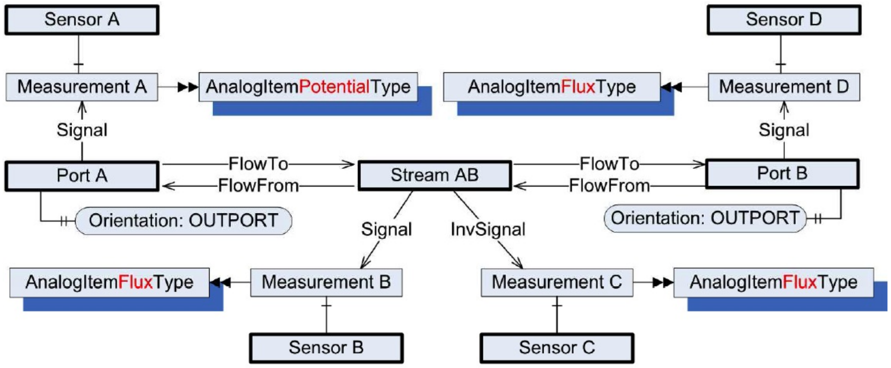

Fig. 9.4 Example of sensors attached to ports and streams

Disturbance Variable, CV – Controlled Variable, etc.) as well as devices topology description. Generally, the L2 model uses object types of L1 model and extends them with additional functionality wherever needed.

9.2.3 Implementation and Results

The information servers have been implemented in the following way:

- L1 aggregation server holds user-defined OPC-UA address space, including user defined user type, methods and references described by standard XML format address space model specified by OPC Foundation.
- It can bind data items to remote servers, working as a client of these servers.
- Upon aggregated data change, it is able to call user defined functions to calculate data from input data item(s) and produce results into output data items. Both input and output data items are represented by OPC-UA variables.
- The chained L2 Server may create subscription to L1 server and monitor data change events on any aggregated or calculated value. Upon that change event each L2 server may perform its own calculations.

Several engineering tools were prototyped to support the whole lifecycle of an OPC-UA information model:

- Information model building tools included the Address Space Model Designer (ASMD), XML editor, and OPC-UA Model Compiler. These tools were used to create the OPC-UA address space, which includes nodes, attributes and their mutual relationships.
- The data binding tool for binding of the data items inside a server address space to external data sources.

- Information model configuration tool for the chained Level 2 servers where it allows to create an instance of a subsystem and define device, topology, and binding views.

The main impact of the consistent use of information models is on selected engineering aspects associated with the implementation of industrial control solutions. One of the most important is the reduced commissioning effort or reduced number of step tests required for setup of an advanced control solution. For instance, for the utility plant illustrated in Fig. 9.2 the cross-layer consistency service allows to build models for all on/off configurations from models of individual devices. This means that only step testing of individuals devices is required, not the step testing of all possible configurations of the utility plant. Assuming significantly simpler models covering individual devices, the overall effort can be reduced down by tens of percent - this is possible by using algorithms based on the structured model order-reduction [14].

9.3 Adaptive Enterprise Energy Management

9.3.1 Problem Description

The provision of fine-grained information and interaction at enterprise-wide level will have a significant impact on future factories and buildings [12], as well as the associated infrastructure such as EV fleets.

By being able to have fine-grained monitoring and control over the enterprise assets, and access to all information, better planning can be realized, while in parallel efficient strategies can be followed realizing the organisation's objectives such as cost efficiency or sustainable operations. In our case, we investigated the benefits of the enterprise as a whole.

We consider the cogeneration plant (as depicted in Fig. 9.1 and page 4) as part of the larger picture which includes the infrastructure available i.e. the company's electric car fleet as well as external energy services such as an energy marketplace. We show that by utilizing the advanced capabilities offered by the energy infrastructure we can realize the vision of more agile enterprises in the future.

This energy management scenario under investigation assumes that the cogeneration utility plant is a source of process steam and electric power for the associated chemical production plant, and additionally, it is also a source of electric power for EV fleet available via the on-site charging stations. The overall goal of the cooperation between the plant and the enterprise energy management system is to cover steam demands and charging needs while following the enterprise's strategic goals. The latter may translate into e.g. minimizing costs, maximizing profit, increase revenue on energy markets etc.

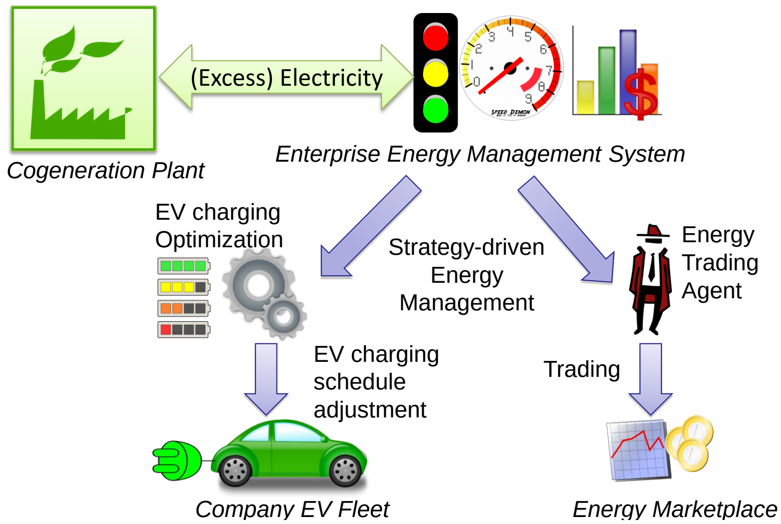

Fig. 9.5 Overview of Adaptive Energy Management with cogeneration plant excess energy availability

9.3.2 Implementation and Results

An overview of the Adaptive Energy Management system is depicted in Fig. 9.5 where we can clearly see the two "tools" available to the enterprise energy management system to achieve its goals i.e. by deciding to charge electric cars and by using software agents to act on its behalf on energy markets. The decision of one or another tool as well as the potential right mix of them may depend on the actual conditions on the cogeneration plant (e.g. excess energy), needs of the EV fleet, current energy prices on the energy marketplace, potential longer term planning e.g. storage of energy etc. Decisions taken may be evaluated also under multiple criteria such as economic benefit, fleet operations, corporate social responsibility goals etc.

We have looked to two scenarios which represent some key cases:

- *Scenario A*: The energy management scenario assumes that the cogeneration power plant is a source of process steam and electric power for the associated chemical production plant, and additionally, it is also a source of electric power for plant owned EV charging stations. The goal of the cooperation is to cover steam demands and charging needs. In this scenario the decision making is done on the power plant which regulates how the excess energy can be used and what percentage is tunnelled to the electric car charging and what goes to the energy provider.
- *Scenario B*: In a similar scenario, but this time interacting with a local energy market [2] as envisioned in Smart Grid era, the excess energy not used in plant can be used to charge the Electric Vehicles. Additionally whatever amount of energy is still remaining after the optimal scheduling of cars has been done, can be

traded on the energy market. Similarly, additional energy that might be needed is also acquired from the energy market. The difference with the previous scenario is that it does not assume interaction with a single stakeholder i.e., the energy provider, and that no adjustments are done on the power plant (hence existing processes remain unchanged). Also the decision making process is now shifted to the orchestrator. This scenario takes advantage of new business opportunities [4], and can be seen as an "add-on" with minimal impact on the power plant operational aspects.

The general workflow is as follows:

- The cogeneration plant is simulated and provides full details on the available excess energy
- A Decision Support System connects to the simulated plant, and acquires the information. Additionally it acquires information about the current EV fleet state and plan, as well as info from the energy marketplace
- After analysis and under the consideration of the enterprise strategies, a decision is taken to (i) store energy by charging the EV fleet, (ii) trade the excess energy on the market, or even a mix among the two that would yield the best benefit e.g. a financial one.
- Upon request the EV charging optimizer undertakes the task to optimally derive a plan and charge existing (and forthcoming) cars on an optimal schedule that coincides with the available to it excess energy.
- Upon request the Energy Trading Agent connects to the market and places the necessary orders to sell the available energy.
- Information on the results of such actions are communicated back to the cogeneration plant and is depicted in the respective enterprise cockpit.

Some assumptions are made here, as well as some extensions to these actions are possible. For instance it is assumed that the cogeneration plant may rely on an external connection to the grid which takes care of potential imbalances. Additionally deviations are also possible; for instance if smaller amounts of energy are needed for the EV optimizer due to dynamic events (e.g. larger than expected cars are now requesting charging etc.) the Energy Trading Agent may issue buy requests to the market to satisfy these needs. Buy requests may also incur also for other reasons, e.g. a fall on the cogeneration plant may result to reduced excess energy being available (than originally predicted) and hence the Energy Trading Agent has "claim back" some of the energy sold which means buying the difference on the market (as one fall-back mechanism if others cannot be realized e.g. cover the difference with a different EV charging schedule).

The implementation has been realized with the following components and technologies:

- A simulator of the cogeneration plant. This is realized in Matlab/Simulink. Access to the information is provided by an OPC-UA server.
- An "Orchestrator" which assumes the responsibilities of the DSS and orchestrates the integration and decision making. The Orchestrator itself consists of

three parts i.e. an OPC-UA client that connects to the Matlab/Simulink and subscribes to the events, a Web service client that connects to the EV optimizer cloud service, and a Web service client that connects to the Energy Trading Agent. Additionally this is the central point for collecting data for future analysis, since it handles the communication with all stakeholders. All of the functionalities related to Web services are developed with the Apache CXF framework that offers RESTful capabilities.

- An EV charging optimizer that optimizes the charging schedule of EVs according to the constraints posed. The EV charging optimizer is realized in Java and runs as a SAP HANA Cloud service. The interfaces it offers are RESTful. The EV charging optimizer considers several dynamic conditions such as production forecast, electricity price, number of expected cars, and tries to find a solution under time constraint (or until it is requested to provide the best solution achieved so far). As our main aim was to demonstrate the easy integration with the IMC-AESOP architecture [5] and external services, we have built upon existing work [10], and extended it for different planning circumstances as well as implemented and deployed in the cloud.
- An Energy Trading Agent and an online marketplace for trading energy at 15 min intervals. All parts here were implemented in Java running as Internet services. We have built upon existing work i.e. adaptations have been made to connect to an existing Energy Services Platform [6] and to the associated marketplace [2] .

The prototype developed as proof of concept has shown that information-driven integration among the various parts of the system can be easily realizable by relying on the IMC-AESOP architecture services [5] and technologies [3]. The usage of Cloud-based services enables the interaction among various stakeholders, and the usage of OPC-UA as well as REST based Web services acted as enablers for cross-layer information flow and dynamic adjustments.

Although the initial two scenarios presented here validated in the simulation the benefits that could be provided to future businesses, by letting them managing in a more sustainable way their resources, real-world trials under realistic conditions will be needed to further validate the tangible benefits against the cost of implementation, operation and maintenance of such a complex infrastructure. However, the latter should also be assessed from a holistic point of view, for all possibilities they might enable for future enterprises.

9.4 Alarm Management

9.4.1 Problem Description

Due to ever-increasing complexity of production processes and a growing number of collected data points at high sampling frequencies, current process control and monitoring systems are evolving from the synchronous scan-based approach towards the asynchronous event-based paradigm. Complex Event Processing (CEP) represents a scalable and efficient means of handling large amounts of data via event-based communication. Service-Oriented Architecture (SOA) represents another paradigm in control and monitoring systems design. This architectural approach overcomes the problem of great complexity and lack of interoperability and adaptability of current systems. Hence, the demonstrator described in this section aims at overcoming challenges in control system design by exploiting the service architecture and the CEP technology in the design of the alarm system.

The alarm system is a critical part of any process control system. It is designed with the objective to aid the human operators to handle abnormal process situations. A major challenge of current control systems lies in flooding the operator with alarms during process upsets (even if the alarm system is well maintained) [11]. Alarm floods are potentially unsafe, since the operator may overlook important alarms or assess the situation wrongly because of stress and information overload. Alarm floods can be mitigated by the use of advanced alarm management techniques, such as alarm load shedding and state-based alarming, which were prototyped within the IMC-AESOP project.

9.4.2 Alarm Handling Techniques

The following advanced alarm processing functions were implemented within the IMC-AESOP project:

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 shut-down). 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 priority or alarm limit). For the automated switching between configurations, the state detection logic must be reliable and must not chatter [1].

Alarm load shedding is a technique that 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 low priorities. The aim of this method is to keep the alarm rate at a manageable level (ideally one alarm per minute) 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.

9.4.3 Architecture

The alarm system architecture developed within the IMC-AESOP project was based on using multiple instances of the CEP engine service and dynamic configuration of the queries executed in this service. Fitting into the context of the SOA-based IMC-AESOP architecture, the following services were implemented to interact with the CEP engine:

- **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).
- **Configuration Service** provides information of the current process state (such as startup, normal operation, shutdown, maintenance, fault, off).
- **Alarm Configuration Service** lists all available alarms with their description, properties and settings, and relation to process units and equipment.

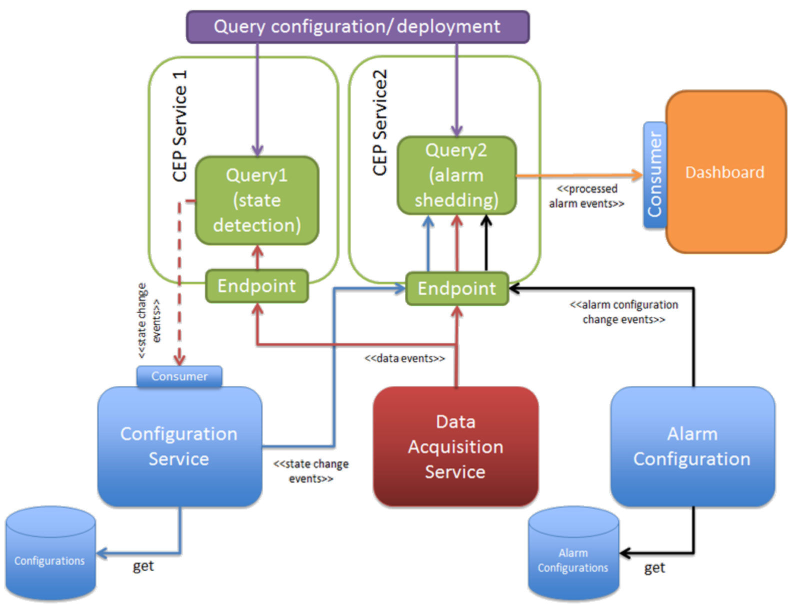

Fig. 9.6 Developed architecture of the alarm system

The alarm processing functionality is provided by two instances of the CEP engine. The first instance receives current measurements of process variables and based on a dynamically configurable query detects the state of the production process. The detected process state is then sent to the Configuration Service. The other CEP engine instance provides the alarm load shedding functionality defined by another configurable query. This engine also provides the state-based alarming functionality by comparing the incoming measurements against the alarm settings corresponding to the current process state.

All interactions among the services depicted in Fig. 9.6 is strictly event-based. All measurements from the Data Acquisition Service are coming to the CEP engines endpoints as events, regardless whether the Data Acquisition Service is implemented as scan-based or event-based. Similarly, the changes of the system state or alarm configuration are also sent as event, which is natively processed by the complex event processing engine.

9.4.4 Simulation Model

A simulation model of a Crude Distillation Unit (CDU) developed in UniSim Design (depicted in Fig. 9.7) was used to measure performance of the new alarm management functions. This model included 131 control system tags sampled with period of 2 seconds. The data included typical abnormal situations resulting in alarm flood conditions.

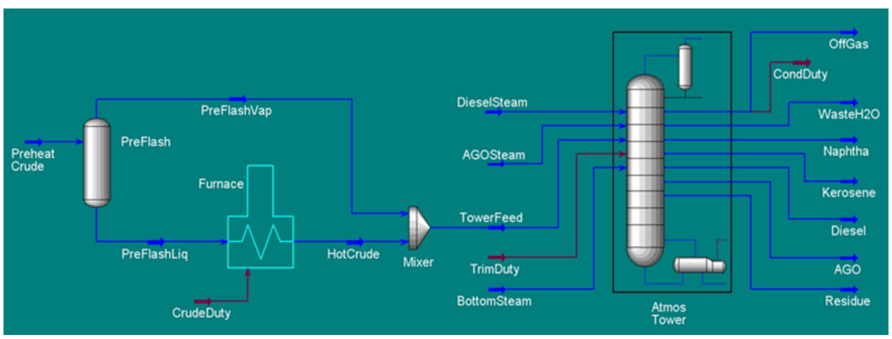

Fig. 9.7 Simulation model of a crude distillation unit

The following states of the CDU were simulated:

- *State 0*: the normal state (the alarm limits design corresponds to light crude oil fed into the column at a medium flow rate)
- *State 1*: light crude oil and a low input flow rate
- *State 2*: light crude oil and a high input flow rate
- *State 3*: heavy crude oil and a high input flow rate

9.4.5 CEP Engine implementation

The CEP engine based on the Microsoft StreamInsight technology was implemented as a Web service using the standard Web services protocol stack, which makes it well usable in heterogeneous systems. The key point in the implementation is identification of messages/events by the "topic" attribute (see Fig. 9.8), which allows to distinguish between different types of messages. The engine allows the definition of the query (containing the actual instructions for event processing) to be flexible and dynamically configurable via the Management API. The actual implementation of the event processing queries is a standard LINQ standing query as used in Microsoft StreamInsight.

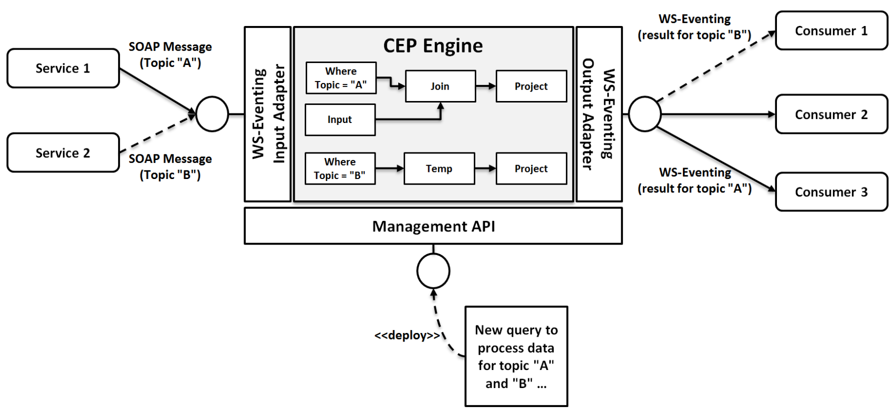

Fig. 9.8 CEP engine as a Web service

9.4.6 Results

Performance of the CEP-based alarm management functions was evaluated using the following alarm system performance metrics.

- Percent of time in flood state – The proportion of time that the operator console is flooded with alarms. The rate at which a single operator is overwhelmed by alarm activations (i.e. when the alarm count per 10 minutes exceeds 10 alarms).
- Average number of alarms in 10 minutes – The alarm rate that the operator is able to efficiently handle in long term is less than 1 alarm in 10 minutes.
- Peak number of alarms in 10 minutes – The maximum rate for the most active 10 minute interval within the evaluated time period is 10 alarms.
- Peak Alarm Minute Rate – The target peak minute rate for the most active minute within the evaluated time period. Target = 2 per minute.

Table 9.1 Alarm performance metrics

	Original state (baseline)	State-based alarming	Alarm load shedding	Target
Percent of time in flood	83%	31%	14%	0%
Average number of alarms in 10 mins	45	8	5	1
Peak number of alarms in 10 mins	78	21	13	10
Peak number of alarms in 1 min	65	15	10	2

As indicated by Table 9.1, state-based alarming significantly improves all the four evaluated alarm system performance metrics. Alarm load shedding in combination with state-based alarming further improves the results by distributing the alarm load more evenly along the time axis. It also slightly reduces the alarm count, since some the low-priority alarms return to normal due to operator actions addressing other alarms.

9.5 Conclusion

The demonstrators described in this chapter were implemented as a combination of several scenarios highlighting advantages of event-driven processing, service orientation and information modelling for improved cross-layer consistency management, adaptive enterprise energy management and alarm management.

Throughput and service availability measurements showed that cross-layer synchronisation implemented based on OPC-UA does not negatively impact the overall performance of data exchange between individual layers. Moreover, quite significant reductions of engineering efforts – and costs – can be realized by systematic adoption of SOA within industrial process control systems.

Also the consistent use of cloud-based services helped to improve interoperability between various applications as well as interactions among various stake-holders in the energy market scenario. OPC-UA as well as REST based Web services enabled cross-layer information flow and dynamic adjustments of schedule allowing to respond to market changes in an agile way.

Finally, in the alarm management scenario, CEP engine was harnessed to support implementation of advanced dynamic alarm handling methods, such as state-based alarming and alarm load shedding. The performance measurement was done primarily on the basis of alarm performance metrics whose values indicated significant reduction of alarm flood condition (by 50–70%) and the peak alarm rates.

Acknowledgment

The authors would like to thank the European Commission for their support, and the partners of the EU FP7 project IMC-AESOP (www.imc-aesop.eu) for the fruitful discussions. We would like also to explicitly thank Ji Hu, Mario Graf, Dejan Ilic, and Per Goncalves Da Silva for their contributions.

References

[1] Hollifield B, Habibi E (2007) Alarm Management: Seven Effective Methods for Optimum Performance. Instrumentation, Systems, and Automation Society
[2] Ilic D, Goncalves Da Silva P, Karnouskos S, Griesemer M (2012) An energy market for trading electricity in smart grid neighbourhoods. In: 6th IEEE International Conference on Digital Ecosystem Technologies – Complex Environment Engineering (IEEE DEST-CEE), Campione d'Italia, Italy
[3] Jammes F, Bony B, Nappey P, Colombo AW, Delsing J, Eliasson J, Kyusakov R, Karnouskos S, Stluka P, Tilly M (2012) Technologies for SOA-based distributed large scale process monitoring and control systems. In: 38th Annual Conference of the IEEE Industrial Electronics Society (IECON 2012), Montréal, Canada.
[4] Karnouskos S (2011) Demand side management via prosumer interactions in a smart city energy marketplace. In: IEEE International Conference on Innovative Smart Grid Technologies (ISGT 2011), Manchester, UK
[5] Karnouskos S, Colombo AW, Bangemann T, Manninen K, Camp R, Tilly M, Stluka P, Jammes F, Delsing J, Eliasson J (2012) A SOA-based architecture for empowering future collaborative cloud-based industrial automation. In: 38th Annual Conference of the IEEE Industrial Electronics Society (IECON 2012), Montréal, Canada.
[6] Karnouskos S, Goncalves Da Silva P, Ilic D (2012) Energy services for the smart grid city. In: 6th IEEE International Conference on Digital Ecosystem Technologies – Complex Environment Engineering (IEEE DEST-CEE), Campione d'Italia, Italy
[7] Karnouskos S, Ilic D, Goncalves Da Silva P (2012) Using flexible energy infrastructures for demand response in a smart grid city. In: The third IEEE PES Innovative Smart Grid Technologies (ISGT) Europe, Berlin, Germany
[8] Mahnke W, Leitner SH, Damm M (2009) OPC Unified Architecture. Springer
[9] OPC Foundation (2011) OPC UA Specification Part 8 – Data Access (RC 1.02)
[10] Ramezani M, Graf M, Vogt H (2011) A simulation environment for smart charging of electric vehicles using a multi-objective evolutionary algorithm. In: First International Conference on Information and Communication on Technology for the Fight against Global Warming (ICT-GLOW 2011), Toulouse, France, August 30-31, Springer Berlin Heidelberg, Lecture Notes in Computer Science, vol 6868, pp 56–63, DOI 10.1007/978-3-642-23447-7_6
[11] Rothenberg D (2009) Alarm Management for Process Control: A Best-practice Guide for Design, Implementation, and Use of Industrial Alarm Systems. Momentum Press

[12] Sauter T, Soucek S, Kastner W, Dietrich D (2011) The evolution of factory and building automation. Industrial Electronics Magazine, IEEE 5(3):35–48, DOI 10.1109/MIE.2011.942175
[13] SmartGrids ETP (2012) SmartGrids SRA 2035 – Strategic Research Agenda. Tech. rep., SmartGrids European Technology Platform, European Commission, URL http://www.smartgrids.eu/documents/sra2035.pdf
[14] Trnka P, Sturk C, Sandberg H, Havlena V, Rehor J (2013) Structured Model Order Reduction of Parallel Models in Feedback. IEEE Transactions on Control Systems Technology 21(3):739–753