Chapter 5
Migration of SCADA/DCS systems to the SOA cloud

Jerker Delsing, Oscar Carlsson, Fredrik Arrigucci, Thomas Bangemann, Christian Hübner, Armando W. Colombo, Philippe Nappey, Bernard Bony, Stamatis Karnouskos, Johan Nessaether, Rumen Kyusakov

Abstract As process control and monitoring systems based on a Service-Oriented Architecture (SOA) are maturing, the need increases for a systematic approach to migrate systems. The legacy systems are traditionally based on a strict hierarchy and in order to gradually allow additional cross-layer interaction, the migration procedure needs to consider both - functionality and architecture of the legacy system. The migration procedure proposed here aims to preserve the functional integration, organize the SOA cloud through grouping of devices, and maintain the performance aspects such as real-time control throughout the whole migration procedure.

5.1 Introduction

In order to include Service-Oriented Architecture (SOA) in the continuous evolution of control and monitoring systems there is a need for a strategy for successive

Jerker Delsing, Rumen Kyusakov
Luleå, Sweden e-mail: jerker.delsing@ltu.se,rumen.kyusakov@ltu.se

Oscar Carlsson, Fredrik Arrigucci, Johan Nessaether
Midroc Electro AB, Sweden e-mail: oscar.carlsson@midroc.com,fredrik.arrigucci@midroc.com, johan.nessaether@midroc.com

Thomas Bangemann, Christian Hübner
ifak, Germany e-mail: thomas.bangemann@ifak.eu,christian.huebner@ifak.eu

Armando W. Colombo
Schneider Electric & University of Applied Sciences Emden/Leer, Germany e-mail: armando.colombo@schneider-electric.com,awcolombo@technik-emden.de

Philippe Nappey, Bernard Bony
Schneider Electric, France e-mail: philippe.nappey@schneider-electric.com,bernard.bony@schneider-electric.com

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

migration of a legacy system into a complete SOA plant. This chapter provides a discussion on how to migrate a system to a Service-Oriented Architecture and how a migration can be expected to affect operations.

This process of migrating from a legacy process control and monitoring system to a Service-Oriented Architecture supports focus on the functionality and the loose coupling of heterogeneous systems to fit dynamic business needs. The legacy systems typically have proprietary protocols and interfaces resulting in vendor lock-ins and possibly site specific solutions; however with SOA these systems can be wrapped, extended or replaced, and integrated in a modern infrastructure.

There is a considerable need to meet various migration requirements for small as well as large scale investments, projects and upgrades of a process control and monitoring system. Here the focus is on migration of large distributed automation systems. The migration towards new functionality, new technology as well as new systems, is risky and therefore the risks of downtime, poor performance and even failure to train personnel must be eliminated. Structured use of risk analysis facilitates the evaluation of different migrations paths.

The migration strategy has its starting point in the business needs, and ideally makes it possible to migrate from a legacy to new system seamlessly without noticeable interruptions at shop-floor and business levels. A migration plan for the pertinent plant, should be compiled, and this has to be validated against global migration plans in order to assure that there are no direct interdependencies with other systems (local and enterprise wide).

It is important to evaluate the migration afterwards and question whether the requirements are fulfilled. Therefore the requirements must be quantifiable and measurable. For example, in order to minimize negative impact of the migration enterprises need measurable requirements for effects like downtime, control problems, costs, interoperability, performance and possibly personnel training. Generic migration strategies, where the different paths and steps are discussed in some more detail, are described hereunder. For this generic migration, the proposed methodology will be developed providing general directions to implement an efficient and low-risk transition from an old system to a SOA based monitoring and control system in process industry environment.

The legacy systems are typically implemented following the hierarchically organized 5-level model as defined within the ISA-95 / IEC 62264 standard (www.isa-95.com). Operations, defined by that standard, are inherent to established production management systems [7]. In this context, concepts for integrating legacy systems, specifically on lower levels, into Service-Oriented Architecture based systems can be seen as business enablers to take the customer from where she/he is today [9] into the future.

Several provider of today's enterprise systems, Level 4 in the ISA-95 architecture (please refer to Chapter 2) already support service-driven interaction e.g. via Web services. Service-Oriented Architecture is an approach used at this level. Services are also used for integration between Level 3 and Level 4 systems, available on the market. OPC-UA [13] is a technology spreading-up to be used. PLCopen in close cooperation with OPC Foundation, defined a OPC-UA Information Model for IEC

61131-3. A mapping of the IEC 61131-3 software model to the OPC-UA information model, leading to a standard way how OPC-UA server-based controllers expose data structures and function blocks to OPC-UA clients like HMIs was defined [1]. OPC-UA relies on Web service based communication. Such activities can be seen as attempts to move towards the use of common technologies across different levels of production systems.

By abstracting from the actual underlying hardware and communication-driven interaction and focusing on the information available via services, the complete system is managed and controlled by service-driven interactions. Services can be dynamically discovered, combined and integrated in mash-up applications. By accessing the isolated information and making the relevant correlations, business services could evolve; acquire not only a detailed view of the interworking of their processes but also take real-time feedback from the real physical-domain services and flexibly interact with them.

The novelty of migrating from a legacy process control system into a SOA, is to do that in a structured way, gradually upgrade highly integrated and vendor-locked standards into a more open structure while maintaining the functionality. The challenges of step-wise migration of a highly integrated vendor-locked DCS and/or SCADA are discussed. From here the necessary migration technology and procedures are proposed. The critical migration technology proposed is based on the mediator concept (as described in Chapter 2). The migration procedure proposed is based on a functionality perspective and comprises four steps: initiation, configuration, data processing and control execution. It is argued that these steps are necessary for the successful migration of DCS and SCADA functionality into a service-based automation cloud.

5.2 Challenges in migrating industrial process control systems

Today's control systems, as used in process or manufacturing automation, are typically structured in an hierarchical manner as illustrated in Chapter 2 in Fig. 2.1.

IEC 62264 (or originally ISA-95) [7] is the international standard for the integration of enterprise and control systems, developed to provide a model that end users, integrators and vendors can use when integrating new applications in the enterprise. The model helps to define boundaries between the different levels of a typical industrial enterprise. ISA-95/IEC 62264 define five levels. For each of these five levels certain problems and challenges become eminent when considering their implementation using a SOA based approach.

Whereas Level 0 is dedicated to the process to be controlled itself, Level 1 connects the control systems to the process by sensors and actuators. Through the sensors the control system can receive information about the process and then regulating the process through the actuators. Sensors convert temperature, pressure, speed, position etc. into either digital or analogue signals. The opposite is done by actuators. Including not only valves but also motors and motor equipment such as frequency converters in actuators, it can be said that the level of installed intelligence varies very much. Legacy implementations use a scan based approach reading and writing data from/to sensors/actuators. Which differs fundamentally to the event based nature of a SOA approach [11, 8]. Migration on Level 1 has to some extent been described [3] with focus on transition from scan-based to SOA event based communication when it comes to analogue signals.

At Level 3, operational management of the production is done, where Manufacturing Execution Systems (MES) provide multiple information and production management capabilities. In the context of control hierarchy, however, its main function is the plant-wide production planning and scheduling. In a continuous process plant, the results of scheduling are used as production targets for individual shifts, and consequently, translated by engineers and operators into individual set points and limits. Level 3 integrates information about production and plant economics and provides detailed overview about the plant performance. If the production is straight forward with few articles and small production site, a dedicated Level 3 system might not bring added value. Some typical MES/MIS functionality is instead put in Level 2 and/or in the ERP-system (Level 4). At Level 4, typically Enterprise Resource Planning Systems (ERP) are installed for strategic planning of the overall plant operation according to business targets. Migration into SOA at Level 3 and 4 does not differ significantly for factory automation and process control systems [2].

At Level 2 there are some non-resolved challenges of migration when it comes to the process industry. Distributed monitoring and control enables plant supervisory control. The distributed control system (DCS) of a large process plant is usually highly integrated compared with a SCADA solution which is standard in factory automation. The SCADA is a supervisory system for HMI and data acquisition and the system communicates through open standard protocols with subordinated PLCs. The PLCs in the SCADA solution are autonomous compared to their counterpart, which sometimes are referred to as controllers, in the DCS. In this paper the process control system is defined as a DCS including HMI workstations, controllers, engineering station and servers all linked by a network infrastructure. A DCS is truly "distributed" with various tasks being carried out in widely dispersed devices. Migration of Level 2 functionality in the form of a DCS exhibits challenges when it comes to co-habitation between legacy and SOA as well as the migration of the control execution [11, 8]. Here the DCS is exemplified by a server/client based system as depicted in Fig. 5.1, which is a common topology.

When migrating the DCS into SOA there are certain requirements based on expectations from business, technical and personnel perspectives:

- The new architecture and the migration strategy must assure the same level of reliability and availability as the legacy system.
- The migration procedure must not induce any increased risk for staff, equipment or process reliability and availability.
- After the migration the plant must still provide the same or a better process, extended service life of plant (process equipment e.g. pumps, vessels, valves), adequate information and alarms depending on department and personnel skill and

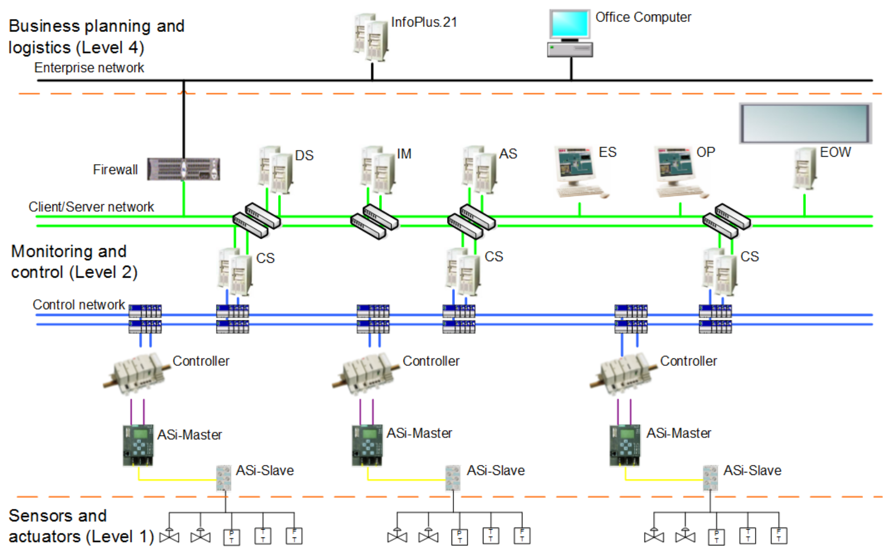

Fig. 5.1 Legacy system architecture

improved vertical (cross-layer) communication with more information available at plant wide level.
- Dynamic changes and reorganisation is expected to be supported, on a continuously running system.
- To handle to co-habitation between the legacy system and the SOA during the migration phase, the SOA solution must support wrapping of legacy sub systems.
- Fieldbus systems, like Profibus PA today already define standardized ways of error indication by devices [5]. With the intelligence built into SOA devices troubleshooting is expected to be improved.

In order to migrate a highly integrated DCS the following challenges should be addressed:

- **Preserve functional integration:** There are advantages with a highly integrated DCS, which give a tight link between the HMI and control execution. Thus design engineering, commissioning and operation can be pursued in a significantly more uniform way. For instance, the HMI and control execution can be configured by the same tools, which facilitates conformity. These advantages must be maintained even though the integration is broken down and substituted by open standards.
- **Grouping of devices:** Within a given system, it must be determined which devices should be migrated to SOA as devices and which devices should be grouped together and the group migrated to SOA. As example a subsystem using feedback and regulation might require legacy interfaces because of real-time demands, therefore such group of devices should be given an SOA interface for the group

using a Mediator and not at device level. This part of the system may be handled as "black-box"

- **Preserve real-time control:** The real time control execution, which in the legacy system is secured in the controllers, must be preserved.

5.3 Functional aspects identified in a DCS

To support the preservation of key functionality during and after the migration certain functional aspects of a generalised Distributed Control System have been identified. In this section a short description is presented for each aspect in order to provide a frame of reference for the migration approaches presented in the following section.

- *Local control loop*: The function of a Local control loop refers to the low-level automated control that regulates a certain part of the plant process, with a relatively low number of actuators and sensors. The control may be continuous or discreet and may use analogue as well as digital actuators and sensors. In many cases the control will require low latency and short sample times, resulting in high bandwidth.
- *Distributed control*: This refers to all forms of control where parts of the control loop are located far away from each other, geographically or architecturally, meaning that the control cannot be executed by a single device (controller) with direct access to both sensors and actuators.
- *Supervisory control*: This form of control is often executed at a higher level based on information from more than one subsystem and is usually much slower than the Local control loop. Often the Supervisory control has no direct access to sensors or actuators but uses aggregated process values as input and actuates through changing the set point of a Local control loop.
- *System aggregation*: Low level devices and subsystems are often presented in an aggregated form to higher level systems in order to show an understandable overview of the system to operators, engineers and others working with the system.
- *Inter-protocol communication*: As different levels of the DCS use different communication standards and protocols all communication between components that are not connected to the same network type and in the same or neighbouring segment need to pass the information through one or more other components. These other components must therefore be able to interpret or translate the information between the different standards and protocols. The effort needed for this kind of communication varies greatly depending on the standards and protocols involved.
- *Data acquisition, display and storage*: Process and system data gathered at all levels of the DCS must ultimately be made available to operators and other connected systems. The availability of correct data is vital to both - operators and

management in order to optimize performance and analyse anomalies. In some cases historical data storage is integrated in the DCS but even in these cases the functionality is not an integral part of the DCS functionality and can be treated as a peripheral system.

- *Alarms and warnings*: All systems have some way of indicating process anomalies to the personnel working with the process. In a well-developed DCS there are many functions related to alarms and warnings that allow distribution of information to the appropriate staff and several modes of suppression and acknowledgement of alarms and warnings.
- *Emergency stop*: The Emergency stop is a vital part of most process control systems, often regulated by national laws and regulations. In a large process plant the emergency stop may be much more complex than simply shutting off the power to all components as this may cause situations where a build-up of heat or pressure, or a chemical reaction would cause a greater disaster than to keep the plant running. It is important that a process control system is able to execute a reliable shut-down procedure even in unexpected situations.
- *Operator manual override*: At most plants it is required that the operator can control parts of the system manually, via an HMI, to handle irregular or unexpected situations. This may be to support maintenance operations where systems are disconnected in a controlled manner or when the operator has to handle unexpected faults in the process or in the automation system.
- *Operator configuration*: Most operator stations allow changing of some parameters in the system such as plant or system operation mode, or control set points for subsystems based on information not available in the automation system.
- *User management and Security*: As many parts of a DCS are interconnected and there are many people with different roles that work with a DCS, it is important that each person is presented with a level of information that is sufficient and relevant for their role. In order to limit human errors as well as malicious actions it is important that all personnel are authenticated for the role in which they are allowed to access the system. The authentication may not always be limited to the software but may instead consist of limiting physical access to certain areas or stations.

5.4 Migration of functionality

In order to ensure and support the preservation of functionality throughout the migration process each functional aspect identified in a DCS have been analysed and for each aspect an example is presented on how the migrated system could provide the functionality in question. These examples are not necessarily the only or the optimal implementations of the functionality but they should provide a sufficient example covering the complete DCS.

5.4.1 Local control loop

At the level of local control loops, the main benefit of applying the SOA communication infrastructure is the richer set of diagnostic and monitoring information that can be delivered and easily integrated into the SCADA systems. By using standard service protocols for the sensor and actuator data delivery, the provisioning stage can be automated to a higher degree than what is possible with the current approaches. Also modifications and upgrades to the system are better supported by using modular, loosely coupled services with support for event-based interactions and resource discovery. As part of the IMC-AESOP project two main approaches are available to migrate the existing control loops to SOA-based solutions proposed by the project:

- For control loops with low real-time requirements (loop times around 100ms or higher), the IMC-AESOP services “Sensory data acquisition” and “Actuator output” can be deployed directly to the embedded sensor/actuator devices. By the use of EXI and CoAP technologies it is possible to provide extensive and non-intrusive diagnostics and monitoring information through wireless links. In many scenarios the achieved efficiency is envisioned to support even the communication of process values via low-bandwidth wireless solutions. Legacy devices supporting firmware updates can be migrated directly to this architecture. For closed black box devices the IMC-AESOP services "Gateway" and "Service Mediator" are required to provide SOA interface and protocol mapping.
- For control loops with strict timing requirements and short loop times (below 100 ms) the direct deployment of “Sensory data acquisition” and “Actuator output” requires deterministic and high-bandwidth PHY/MAC layers such as Industrial Ethernet solutions. For low-bandwidth links, e.g. (Wireless) HART, would likely require gateway/mediator wrapping to migrate the low-level real-time protocols used for the loops with a SOA-ready interface. Thus, simple and time-critical sensors/actuators part of real-time control loops are not migrated to SOA but rather wrapped on a higher level.

5.4.2 Distributed control

A service architecture supports the distribution of the treatments on several systems or devices. As far as it is possible the control is located at the lowest level so that the treatments can be more appropriate due to the knowledge of the local context. Moreover, the amount of data needed to communicate to the upper levels can be reduced.

Intelligence of the control is pushed down in the devices so that treatments remaining at controller level may be performed in other kinds of devices than controllers, like for example network infrastructure devices. Part of the control can be temporarily disconnected without affecting the complete equipment, either for normal replacement or even for upgrading the functionalities. Configurations containing the control logic are stored in a central repository so that exchange of devices is possible without manual reprogramming. A Service Bus middleware can typically support such a decentralized control:

- The components of the Service Bus may be physically distributed on:
  - Existing devices of the system (case of the Fig. 5.2)
  - Existing infrastructure devices like gateways
  - Dedicated devices
- Some of the devices have their own local logic so that they can expose high level services.
- Other devices cannot expose such SOA services. They are either legacy devices or small devices that do not support local logic. Thanks to the Service Bus these devices can nevertheless interoperate in the system.
- The remaining logic required to make the global control of the system is distributed within the Service Bus, i.e. in this example on the two devices supporting a Service Bus component.

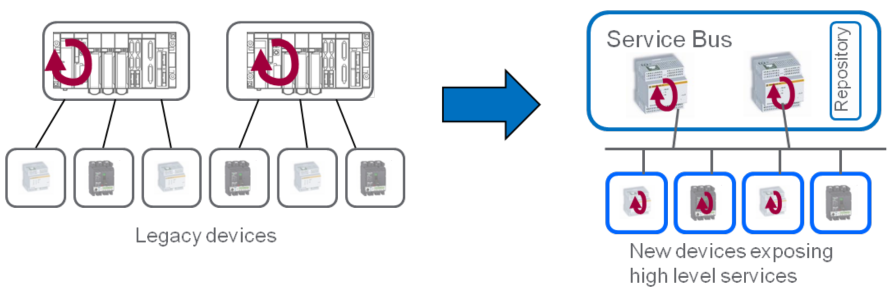

Fig. 5.2 Distributed control with a Service Bus middleware

5.4.3 Supervisory control

In a SOA approach devices can expose directly their data to the other systems at different levels; there is no more a hierarchical structure where device data are first collected by controllers which then feed the supervisory control system. The visibility of the devices is then improved without additional workload. Maintenance and evolutions of the supervisory application are also decoupled from other underlying systems like controllers or OPC servers.

Supervisory control systems can also propose a richer interface while their development is easier thanks to the usage of tools understanding the standard interfaces exposed by the controllers and the devices. These interfaces are typically described through WSDL files.

OPC-UA provides additionally a feature known as programming against type definitions (see Fig. 5.3 below). The principle is that an OPC-UA server supports the definition of complex object types which can be recognized by a client application like a supervisory control. In the server address space both the object type and the object instances are exposed. The supervisory control either already knows the object types exposed by the server or discovers them during the engineering phase. In both cases, the treatments concerning each object instance is programmed only once due to the knowledge of the object type. In this way, supervisory control applications can be quickly developed with libraries of components corresponding to standard object types.

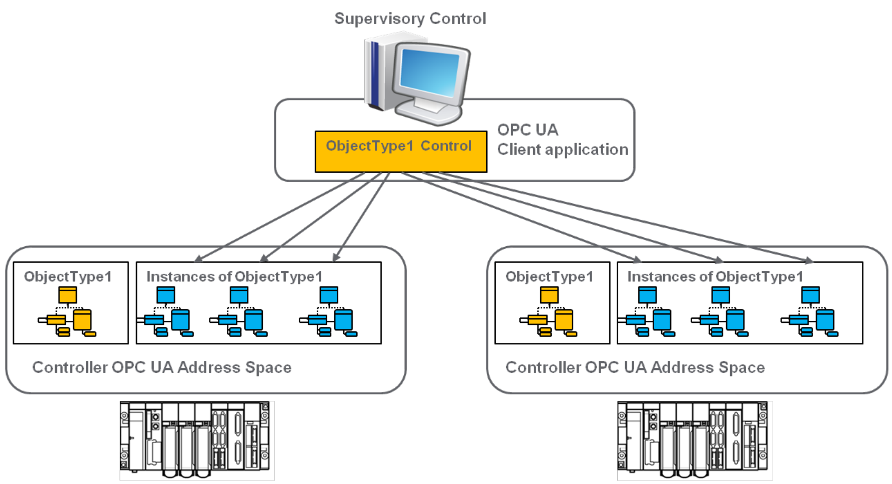

Fig. 5.3 Programming against object types with OPC-UA

5.4.4 System aggregation

As indicated in Fig. 5.4, process plants are separated into several sections. Depending on the nature of the process represented by a section, control can be realized in an encapsulated, but coordinated by master control, manner. This is even more the case in batch applications than in continuous processes. Batch control is a more flexible way for mastering market demands of producing small quantities of changing products (chemical, petrochemical, medical, etc.) at the same production site. Here production equipment like boiler, heat exchanger, distillation colon, or alike, are dynamically combined and controlled according to recipe needs. Support functions like air compression for auxiliary energy provision or cooling aggregates are normally built as package unit also having own controls.

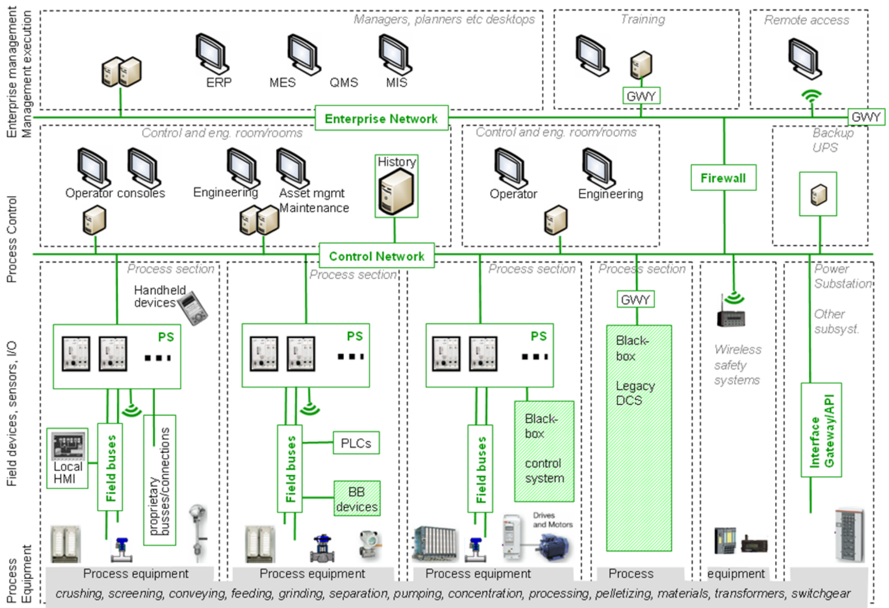

Fig. 5.4 General architecture of a process control system

As it can be seen, today's classical process plants, and associated automation systems are already, even if partially, characterized by

- Aggregation of information dedicated to specific plant sections
- Individual engineering and control of those sections (black boxes)
- Hierarchical engineering concepts for overall/master control
- Supervision down to black box level

Additionally, one can start from the process level to identify plant sections, e.g. performing individual control loops or contributing information to dedicated aspects (like Maintenance) of a plant view, to define data related to each other. Those relations may guide to the definition of application related services, contributing to the SOA. Some elementary services are already defined in the IMC-AESOP architecture [12].

According to the step-by-step Migration approach, those typical representations can be seen as starting point for a specific migration step supporting dedicated integration technologies. The overall migration process will be a series of individual migration steps [3].

Integrating and aggregating data for that purpose requires, knowledge about access path and methods to data as well as syntax and semantics of the data accessed. This information may be derived from project documentation provided by the vendor with the delivery of the control equipment for a plant section or from pre-established knowledge in case of conformance to well established standards. There are well established standards targeting information management at different levels of the ISA-95 layered model, exploitable for integration tasks, like:

- ISA-95
- S 88
- Device Profiles

Use of standard conformant equipment is highly recommendable, as they ensure reimbursement of investments. Afterwards, a summary of a concept for aggregation of data and definition of services at Level 1 and 2 is given continuing work started in SOCRADES [2, 14].

A Gateway and Mediator concept [10] has been introduced as being suitable for realizing integration tasks within the IMC-AESOP framework. This concept supports representing single resources (like a legacy device) to a SOA based environment as well as aggregating and mediating data from a single or multiple resources.

5.4.5 Inter-protocol communication

Interoperability of applications requires fundamental communication capabilities, even if applications are running on inhomogeneous communication platforms. That, in fact, is the usual case for integration tasks. Two, or more, communication channels have to be mapped to each other considering the different characteristics at all protocol layers. There are different approaches known from literature:

- Bridge
- Gateway
- Router

Considering introduction of SOA into process control environment, one will be faced with integration tasks of different type of communication ($4\ldots20mA$ standard wired signals, HART protocol, fieldbus protocols like Profibus PA and others). All to be mapped to a single protocol, as agreed to be used for communication within the SOA.

Within the IMC-AESOP approach Gateway or Mediator concepts are used for protocol mapping, covering interfacing of different protocols, interpreting syntax and semantics of data operated at each communication channel (possibly in a different way) and mapping data to an internal data model of the integration components. The Web Server (interface to SOA) accesses the internal data model and maps the data to an appropriate Web service, conformant to the IMC-AESOP architecture definition [12].

Configuring this mapping is a multi-step approach, while doing configuration for each of the individual communication channels, instantiating an internal object model, representing the targeted view of the underlying system, and defining the mapping roles to the Web services. Knowledge is needed for all the protocols and applications targeted.

5.4.6 Data acquisition, display and storage

Data acquisition, in terms of the current state of the art has many possible solutions and implementation, most commonly using a PLC or some sort of RTU connected to a fieldbus to transfer data as required. In terms of the IMC-AESOP architecture the main objective is to change, or migrate from this kind of traditional systems to smart embedded devices capable of both acquiring the necessary data and encapsulating it in Web services that can later be consumed by any interested party. An example of this migration could be taken from Use Case 2 Oil Lubrication of the IMC-AESOP project.

At the lowest level this use case requires computers capable of calculating flow rates from positive displacement flow meters. These volumetric flow meters generate pulses at frequencies ranging from $1-500Hz$ depending on the model being used. Any conventional PLC or RTU unit has inputs that can detect a frequency of roughly $50Hz$. While this is good enough for certain flow meters it is not nearly enough to cover the whole range of possibilities. There are two possible solutions to this migration problem:

1. One possible solution is to use a legacy flow computer with legacy communication capabilities i.e. modbus. This would enable the flow computer to do the high frequency calculations necessary and transfer the data to a modbus register that could be read by any WS-capable device. The data would then be processed from pulses into flow rate and encapsulated into a WS-Event, or message depending on whatever the requirements are.
2. Another possible solution is to have a fast counter card or specialized inputs integrated to a WS-Capable device. This would imply that the device would both have to be capable of counting, pre-processing and calculating flow rates without any external help. Then it would only be a matter of encapsulating the data in WS form in order to make it available to any interested parties.

Whichever solution is chosen however, it is important to keep in mind that legacy flow metering computers, while limited, have well-defined, well-tested algorithms that calculate flow rate. In the case of migration it is necessary to evaluate whether accuracy or scalability and easy access are more important. The common in both solutions however, is the need for WS encapsulation, which would imply exactly the same work in both cases. It would be necessary to design the corresponding WSDL file so that the device capturing the information could be discovered and subscribed to. Although this might depend on how the WS-Capable device is designed to work.

5.4.7 Alarms and warnings

Alarms can be raised at different levels, either directly by the devices or by upper level systems, processing various information coming from one or several sources.

Additionally to the definition of standardized interfaces defining the content of the alarms, an SOA approach proposes communication mechanisms insuring that:

- The right information will reach the right person in the plant and with an appropriate level of details
- The communication network of the plant will not be overcrowded by useless data

These two goals are achieved by filtering and routing mechanisms, implemented typically by a Complex Event Processing (CEP) technology as investigated in the IMC-AESOP Project.

For the end user, the benefit of a SOA approach is that he will receive only the needed alarms and warnings. The content of the alarms will be filtered depending on the user who is logged into the system, giving the information just required for the actions of the user. For example an operator will be informed that the process is stopped without any further detail while a maintenance team will receive details about the machine breakdown.

5.4.8 Emergency stop

Detection of abnormal conditions requiring an emergency stop can be performed at various levels. Additionally to emergency stop buttons at shop-floor level, the events raised in the different layers of the system can indirectly inform the operators of critical alarms, typically within the supervisory control system. Moreover, complex event processing systems can correlate the information coming from different sources located in any location of the system in order to raise such emergency alarms.

Once the operator has pressed physically a shop-floor button or has selected the emergency stop in an HMI, the equipment must shut-down in a controlled manner which depends on the exact state and topology of the system. The agility of a SOA infrastructure allows managing several shut-down strategies depending on the various emergency conditions as well as adapting these strategies all along the life-cycle of the equipment.

In some context, typically for regulation purpose, the shut-down of the equipment must be done in a given time frame and with a precise sequence of operations. In those cases, safety protocols solutions must be used to manage these particular constraints. There are currently different add-ons existing for classical fieldbuses but for the envisioned systems where IP protocol over Ethernet is largely used, safety solutions based on Ethernet must be carefully considered.

5.4.9 Operator manual override

The devices expose standardized interfaces so that a unique or at least a limited set of tools can be used by the operators for taking the control locally. Then the operators can be well trained and efficient, what is particularly important when an unexpected situation happens, which is a typical case where manual override is required.

The parts of the system where operators have overridden the automatic control must be easily pointed out in the upper levels applications, even if this is a scheduled maintenance where a part of the system is disconnected intentionally. SOA makes possible a direct connection between the upper level and the devices so that such critical information is easily available. Such information is used not only by the operator but also by the upper level applications to reconfigure themselves.

Thanks to the loose-coupling of the SOA approach, most applications at level 2 or level 3 will continue interacting with the manually controlled part of the system without considering its operating mode. Only applications interested by the operating mode will be informed, typically by alarms and events mechanisms.

5.4.10 Operator configuration

The devices expose standardized configuration services so that here also a limited set of tools can be used for local configuration. Then operators have not to get a lot of different tools and to be trained for them. The changes made in the device configuration must be then pushed to the configuration repository in order that after replacement of a device, the same configuration can be downloaded to the new one. Different strategies can be used here, either the operator decides explicitly that the new settings are valid and initiates the backup manually, or the device configuration may be compared periodically to the reference, which is updated if the actual device configuration is different but valid.

The Fig. 5.5 below describes a system where a standard service DeviceManagement is supported by an IMC-AESOP device. A local configuration tool can be used to perform the following actions:

1. Get the current configuration of the device. The response of the GetConfiguration operation is defined with a very generic format. Virtually any kind of device configuration can be retrieved.
2. The operator edits the device configuration with the configuration tool HMI.
3. The tool uploads the new configuration in the device (SetConfiguration operation).
4. Optionally the new configuration is pushed in the configuration repository. This repository will be used in particular in case of device replacement.

Notes:

- In this example the configuration repository is managed within the Service Bus introduced in chapter 4

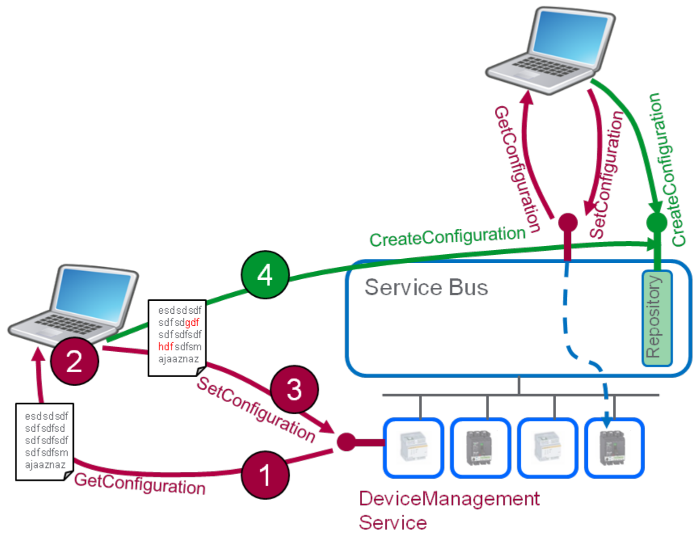

Fig. 5.5 Operator local configuration with a Service Bus

- The right side of the Figure 4 demonstrates that the Service Bus can provide a service view also for legacy devices. It translates legacy protocols and legacy data formats so that it can expose the DeviceManagement service on behalf of the legacy devices.

5.4.11 User management and security

Up to now, a predominant behaviour was to have locally authenticated users (e.g., on-device or department) and devices (if at all). However, this practice created "islands" within the infrastructure that were difficult to be controlled e.g. if they adhere to the corporate policies, and are costly to maintain. However in the IMC-AESOP vision, the security framework should be company-wide and the “visibility” of devices in the cloud makes it easier to have a system-wide view. The migration however towards this infrastructure, will require a lengthy transition process and potentially significant effort to reassess security and risk relevant aspects, test configuration and impact, and move towards integrated management of both users and their rules.

5.5 Migration procedure

Interfacing and integrating legacy and SOA components of a DCS/SCADA system will require some, for the purpose developed and/or adapted, technology. Such integration may be based on some kind of integration component like Gateway or Mediator. Such Gateway or Mediator has the task to bridge the communication from major standardized protocols used close to field applications today: HART communication, Profibus PA in combination with Profibus DP, Foundation Fieldbus, etc. These protocols follow specific characteristics. Some commonalities can be monitored like concepts for device descriptions or integration mechanisms into DCS (e.g. EDD, FDT, FDI). The same bridging task exists regarding communication to higher level, technologies related to Enterprise Application Integration (EAI) or Enterprise Service Bus (ESB) or OPC (OPC DA, OPC-UA) are used having their own characteristics and configuration rules.

The use of Gateways or Mediators is a well proven concept for integrating/connecting and migrating devices, attached to different networks. It is used to transform protocols as well as syntax of data. Semantic integration is hard to achieve. Nevertheless it is possible to do transformation between data centric approaches, as typically followed by fieldbus concepts, and service oriented, event centric, approaches.

The Mediator [10] concept used here is built on the basis of the Gateway concept by adding additional functionality. Originally meant to aggregate various data sources (e.g. databases, log files, etc.), the Mediators components evolved with the advent of Enterprise Service Buses (ESBs) [6]. Now a Mediator is used to aggregate various non WS-enabled devices or even services in SOA [10].

Using Mediators instead of a Gateways, provides the advantage of introducing some semantics or to do pre-processing of data coming from legacy networks, e.g. representing a package unit. Due to the diversity of data, or different aspects of interest, that different applications request different types (e.g. quality, quantity and granularity) of data, interface devices will normally be built as a combination of Gateway and Mediator. As it may also be applicable to integrate service oriented sections (e.g. retro-fit of a plant section or replacement of a package unit) into existing systems, this Gateway and Mediator concept can be extended to represent services into data centric systems (today's legacy systems). Mediator as well as Gateway concepts, both are powerful means for integrating single legacy devices or legacy systems encapsulating "isolated" functionalities.

Whereas the operational phase of a system will benefit from the functionalities described above from the beginning of the migration process, engineering will be characterized by a step-wise approach, starting with defining services representing the legacy device or system, followed by separate engineering steps for the legacy part and the SOA based part using those services defined. Specific configuration effort for the Mediator or Gateway itself is needed. It is advisable, that commissioning will also be done in a multiple step approach, starting at the isolated components followed by their integration into the overall system.

Considering the layout of a server/client-based SCADA/DCS a stepwise migration through four major steps is proposed. The four major steps may contain substeps and may be spread out over a long period of time but each major step should be completed before the following step is initiated. The four major steps suggested are:

- Initiation

- Configuration
- Data processing
- Control execution

During the whole migration the system will require one or more mediators to allow communication between the SOA components and the parts of the legacy system that not yet has been migrated. The propagation of the mediator and the growth of the SOA cloud are exemplary applied to the migration of the legacy SCADA/DCS presented in Fig. 5.1. Making emphasis in the DCS-part, the set of Fig. 5.6 up to Fig. 5.9 shows the different results reached throughout the whole migration process.

5.5.1 Step 1: Initiation

The initial SOA "cloud" needs some of the basic services presented in [12] in order to support basic communication and management of the cloud. Once the basic architecture is constructed the first peripheral subsystems can be migrated and new components can be integrated in SOA. In migration of subsystems, as well as integration of new components, some consideration must be made of the limitations of the mediator and its communication paths.

The systems migrated in this step include sub system which are not directly part of the highly integrated DCS:

- Low level black box
- High level systems for business planning and logistics such as maintenance systems

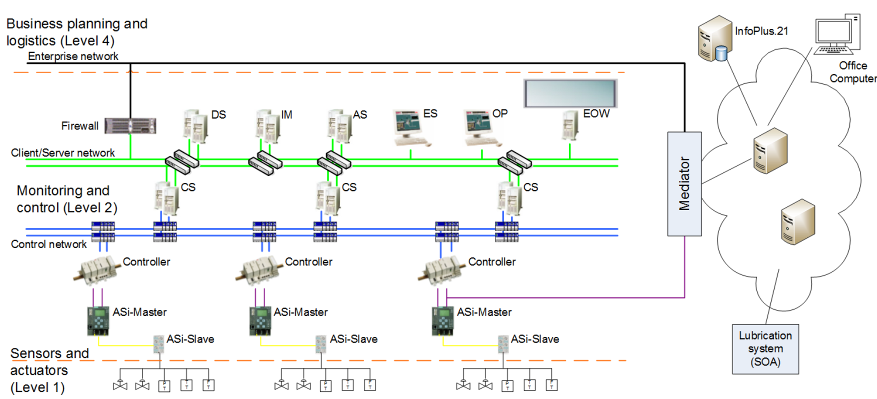

Fig. 5.6 DCS after the first step of migration

Migration is limited to the operational phase of the systems integrated. Within that step, engineering is out of the scope of migration. An appropriate engineering approach, dedicated to this migration step, is doing multi-step configuration:

- Configuration of every legacy system including the legacy interface within the mediator
- Configuring the SOA system
- Configuring the model mapping within the mediator

Exploiting machine readable legacy configuration information would be helpful for every step. Today, configuration information is available through different technologies e.g. GSD, EDD, paper documentation. This type of information is mostly available for single devices. Engineering stations take these information as input and generate system configuration information in proprietary formats.

At this point several parts of the functional aspects can be considered to be at least partially migrated. Most likely some of the Local control loop functionality is migrated. Inter-protocol communication is required both in the migrated and the traditional parts of the system and user management and security must be at least partially implemented in the SOA-system without compromising existing security or creating unnecessary obstacles for users or user administrators. System aggregation, emergency stop, alarms and warnings, operator manual override and operator configuration have all been implemented in the SOA-system to the extent required by the migrated subsystems, while the respective functionality in the traditional system is virtually untouched.

5.5.2 Step 2: Configuration

This is the first step where components that are heavily integrated in the DCS are migrated. The purpose of this step is to migrate parts of the DCS that do not require very short response times or the regular transport of large amounts of data. Please refer to Fig. 5.7. The majority of functions that qualify for this migration step are in some way concerned with configuration of different parts of the DCS. The point of origin for most, if not all, configuration is the Engineering Stations (ES) which is used for engineering and configuration of most parts of the DCS.

As the ES is migrated to SOA, this constitutes a major increase in the number of services the Mediator needs to supply to the SOA cloud as it must in addition to the operational data migrated in the first step represent configuration aspects of all legacy systems and devices not yet migrated, and allow configuration of all systems and devices. This means that configuration of low-level devices and control is done on the ES in a SOA environment using configuration services provided by the mediator, the configuration is then compiled by the mediator into their respective legacy formats and downloaded into the legacy controllers.

Configuration of HMI, Faceplates and associated systems is similarly done in SOA and converted by the mediator to a format that can be downloaded into the legacy Aspect servers and other legacy systems. The configuration of legacy devices from SOA might also require that the mediator is able to extract legacy designs and configurations that may be stored in aspect servers or controllers can be reused and modified by the SOA Engineering stations.

This approach may be combined with doing multi-step configuration described in the former step.

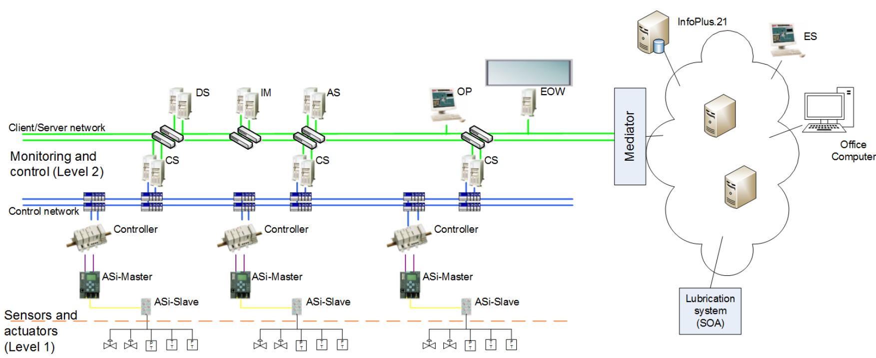

Fig. 5.7 DCS after the second step of migration

As legacy systems usually do not provide sufficient meta-data, sufficient configuration information can not necessarily be extracted by a Mediator from the installation (legacy systems). Consequently, for overall engineering a SOA engineering station should being able to import relevant configuration information of different legacy systems in addition to the limited capabilities provided by the Mediator itself. If such a tool would be available, one could design a mediator acting as configuration station for different legacy systems (compile configuration information into legacy formats) while receiving basic configuration information from the SOA engineering station.

As most of the functionality of everyday operation should be unaffected by the migration of engineering and configuration tools, only a few of the functional aspects are affected. Most notably there will be an increased need for inter-protocol communication and there may be a possibility to utilize more of the functionality described in Supervisory control. In addition the migration of the Engineering station means that some additional parts of user management and security is migrated, but apart from those, most functional aspects should be similar to that those of the first step in the migration procedure.

5.5.3 Step 3: Data processing

In this third step, the migration includes all components and/or subsystems that do not require short response time (millisecond range) not currently achievable by the

SOA technology (refer to Fig. 5.8). This includes Operator Clients (OP) and Operator Overview Clients (EOW) as well as Aspect Servers (AS) and Information Management Servers (IM). As all points of user interaction with the system is now moved to SOA this means that the legacy Domain Servers (DS) are redundant. However, as user management and security needs to be available in SOA from the first step of the migration, there is probably no need for the Domain Servers in the SOA cloud, although the functionality can be considered to be migrated.

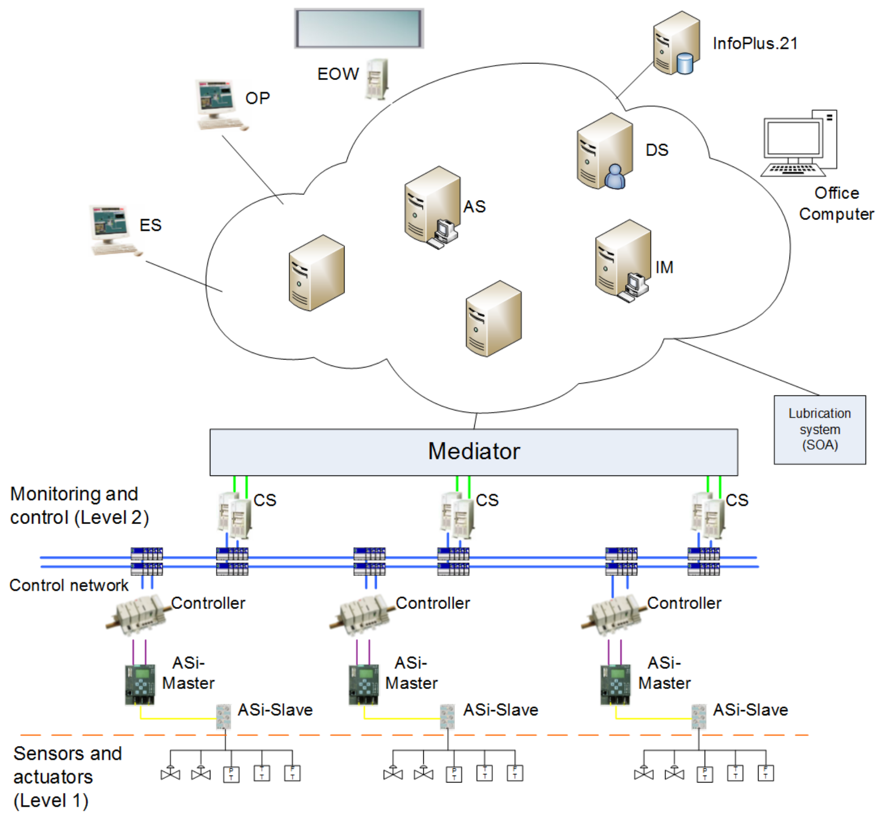

Fig. 5.8 DCS after the third step of migration

The migration the Operator Clients and the Aspect and Information Management Servers mean that the role of the mediator is once again fundamentally changed. In Step 3 of the migration there is less of a need for a flexible mediator that can communicate with a lot of different legacy components, the new requirements are more concerned with a need to present large amounts of data available from legacy controllers to the migrated Operator Clients and other data processors and consumers. This activity is closely related to the purpose of the Connectivity Servers (CS) and it is suggested that the mediator in Step 3 is implemented as a new interface in the Connectivity Servers.

At this stage several operator-centric parts of the functionality are completely migrated. Most significantly Operator manual override and Operator configuration are fully migrated. All of Data acquisition, display and storage, except the first level of acquisition of data from the devices up to the controllers, are also migrated at this step. As the functionality for data acquisition is migrated some additional functionality for System aggregation might be required to present the data from underlying systems in the case where this is not sufficiently covered by the traditional systems. In addition all of the Alarms and warnings functionality, apart from some generation of alarms at the controller level, is migrated and so is most of User management and security.

5.5.4 Step 4: Control execution

In the fourth and final step of migration the time has come to migrate the functionality traditionally provided by controllers (shown in Fig. 5.9). As control execution in the legacy system can be grouped together with several control functions in one controller, or in some cases spread out with different parts of a control function executed by more than one controller, it is of utmost importance that control execution is migrated function by function rather than controller by controller.

Depending on the performance requirements of each control function there may be a need for different strategies for different functions. In the cases where SOA compliant hardware is available for all functions an Active Migration may be suitable where a detailed schedule can be made over the migration of all functions, enabling a controlled migration towards a set deadline. In other cases it may be suitable to allow legacy controllers to fade out as functions are migrated in the course of normal maintenance and lifecycle management of the plant. The fade out option means that Step 4 of the migration may take a very long time but it may save costs as legacy devices are used for their full lifetime, while most benefits of SOA are already available.

During this fourth step most of the functionality migrated relates to control at some level, as most of the monitoring, engineering and administration already has been moved to the SOA-system. In particular this relates to Local control loop, Distributed control and Supervisory control. Another key function that is migrated in this step is the Emergency stop, which can be considered a form of human-in-the-loop control with some very specific conditions. As each specific control function is migrated so are the related support functions such as System aggregation, Data acquisition, display and storage and Alarms and warnings.

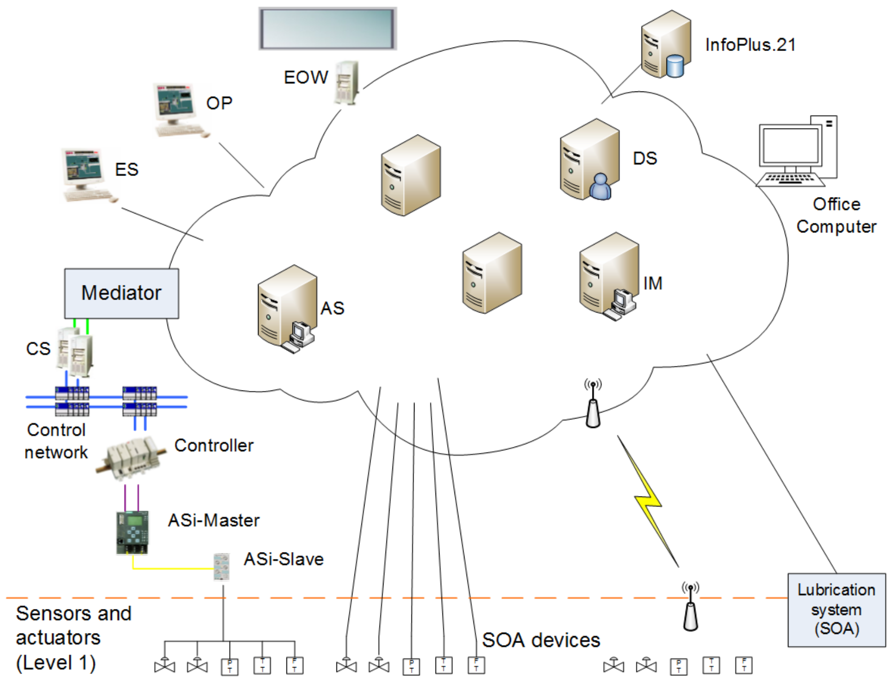

Fig. 5.9 DCS after the forth step of migration

5.6 Conclusion

Following on and extending the initial migration concepts introduced in [3] and further detailed in [4], the novelty of migrating from a traditional hierarchical ISA95-based legacy process control system into a SOA-compliant ISA-95-based process control system is to proceed in a structured way, gradually upgrading highly integrated and vendor-locked standards into a more open structure while maintaining the functionality. Note: The migration concept presented here is not modifying the structural hierarchy of an ISA-95-based process control system but allowing it to functionally behave as a highly distributed flat architecture based on services located on physical components and/or on the cyber-space represented by a service-cloud.

A procedure for migrating the functionality of a DCS/SCADA to a cloud SOA based implementation is proposed. The procedure comprises 4 distinct steps and makes use of mediator technology. These 4 steps are designed to maintain the feeling of conformity between HMI and control execution and to ensure that the target system exhibits full transparency and supports open standards.

The migration procedure is further analysed through a breakdown of the functionality of a DCS/SCADA and how the functionality can be migrated to SOA. A short description of an exemplifying proposed implementation for each functional aspect is provided and Table 5.1 provides a summary of how these functional aspects

Table 5.1 Functional aspects mapped to Migration steps

Functional Aspect	Step 1	Step 2	Step 3	Step 4
Inter-protocol communication	∗	◯		
User management and security	∗	∗	◯	
Operator manual override	(∗)		◯	
Operator configuration	(∗)		◯	
System aggregation	(∗)		(∗)	(∗)
Data acquisition, display and storage			◯	∗
Alarms and warnings			◯	∗
Local control loop	∗			◯
Emergency stop	(∗)			◯
Supervisory control		(∗)		◯
Distributed control				◯

are related to each migration step. Many aspects are partially migrated (indicated by '∗') or can be migrated depending on the scenario (indicated by '(∗)') at different steps of the migration while there is a certain step where the main part of the functionality is migrated (indicated by '◯').

Using this step-wise approach, utilizing SOA and mediator technology, it is argued that the SOA approach will preserve functional integration, support grouping of devices, preserve real-time control and successfully address safety loops. With an emphasis on the DCS-part of an exemplifying legacy control system, the authors applied the approach and present the results reached.

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.

References

[1] Bohn H, Bobek A, Golatowski F (2006) Sirena - service infrastructure for real-time embedded networked devices: A service oriented framework for different domains. In: Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on, pp 43–43, DOI 10.1109/ICNICONSMCL.2006.196

[2] Colombo AW, Karnouskos S (2009) Towards the factory of the future: A service-oriented cross-layer infrastructure. In: ICT Shaping the World: A Sci-

entific View, vol 65-81, European Telecommunications Standards Institute (ETSI), John Wiley and Sons

[3] Delsing J, Eliasson J, Kyusakov R, Colombo AW, Jammes F, Nessaether J, Karnouskos S, Diedrich C (2011) 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.

[4] Delsing J, Rosenqvist F, Carlsson O, Colombo AW, Bangemann T (2012) Migration of industrial process control systems into service oriented architecture. In: 38th Annual Conference of the IEEE Industrial Electronics Society (IECON 2012), Montréal, Canada.

[5] Diedrich C, Bangemann T (2007) PROFIBUS PA Instrumentation Technology for the Process Industry. ISBN-13 978-3-8356-3125-0, Oldenbourg Industrieverlag GmbH

[6] Hérault C, Thomas G, Lalanda P (2005) Mediation and enterprise service bus: A position paper. In: Proceedings of the First International Workshop on Mediation in Semantic Web Services (MEDIATE), CEUR Workshop Proceedings, URL http://ceur-ws.org/Vol-168/MEDIATE2005-paper5.pdf

[7] IEC (2007) Enterprise-control system integration –Part 3: Activity models of manufacturing operations management (IEC 62264-3)

[8] Jammes F, Smit H (2005) Service-oriented paradigms in industrial automation. Industrial Informatics, IEEE Transactions on 1(1):62–70, DOI 10.1109/TII.2005.844419

[9] Karnouskos S, Colombo AW (2011) Architecting the next generation of service-based SCADA/DCS system of systems. In: 37th Annual Conference of the IEEE Industrial Electronics Society (IECON 2011), Melbourne, Australia.

[10] Karnouskos S, Bangemann T, Diedrich C (2009) Integration of Legacy Devices in the Future SOA-based Factory. In: 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM), Moscow, Russia.

[11] Karnouskos S, Colombo AW, Jammes F, Delsing J, Bangemann T (2010) Towards an architecture for service-oriented process monitoring and control. In: 36th Annual Conference of the IEEE Industrial Electronics Society (IECON-2010), Phoenix, AZ.

[12] 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.

[13] Mahnke W, Leitner SH, Damm M (2009) OPC Unified Architecture. Springer

[14] Taisch M, Colombo AW, Karnouskos S, Cannata A (2009) SOCRADES Roadmap: The future of SOA-based factory Automation