Chapter 2
State of the Art in Industrial Automation

Thomas Bangemann, Stamatis Karnouskos, Roberto Camp, Oscar Carlsson, Matthias Riedl, Stuart McLeod, Robert Harrison, Armando W. Colombo and Petr Stluka

Abstract In the last decades, industrial automation has become a driving force in all production systems. Technologies and architectures have emerged alongside the growing organisational structures of production plants. Every innovation had to start

T. Bangemann (✉) · M. Riedl
ifak, Magdeburg, Germany
e-mail: thomas.bangemann@ifak.eu

M. Riedl
e-mail: matthias.riedl@ifak.eu

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

R. Camp
FluidHouse, Jyväskylä, Finland
e-mail: roberto.camp@fluidhouse.fi

O. Carlsson
Midroc Electro AB, Stockholm, Sweden
e-mail: oscar.carlsson@midroc.com

S. McLeod · R. Harrison
University of Warwick, Coventry, UK
e-mail: c.s.mcleod@warwick.ac.uk

R. Harrison
e-mail: robert.harrison@warwick.ac.uk

A. W. Colombo
Schneider Electric, Marktheidenfeld, Germany
e-mail: armando.colombo@schneider-electric.com

A. W. Colombo
University of Applied Sciences Emden/Leer, Emden, Germany
e-mail: awcolombo@technik-emden.de

P. Stluka
Honeywell, Prague, Czech Republic
e-mail: petr.stluka@honeywell.com

A. W. Colombo et al. (eds.), *Industrial Cloud-Based Cyber-Physical Systems*,
DOI: 10.1007/978-3-319-05624-1_2,

from the latest state-of-the-art systems within the respective domain. While investigating the introduction of service-oriented architectures to automation, and even down to the shop floor, one has to consider latest standards, proofed technologies, industrial solutions and latest research works in the automation domain. This chapter tries, without any claim to completeness, to provide a short summary of today's situation and trends in automation.

2.1 Architecture of Production Systems

Several efforts to date have been directed towards defining structural and architectural aspects of production management systems. The most popular and applied in practice are the definitions set up within the ISA-95/IEC 62264 [21] standard. Typically, today's production systems (factory and process) are structured into a five-level hierarchical model (as depicted in Fig. 2.1). Besides this hierarchical, well-known model, IEC 62264 defines a manufacturing operations management model (like production control, production scheduling, maintenance management, quality assurance, etc.), which is not as popular, but implicitly represented by real installations.

The standard defines functions mainly associated to levels 3 and 4, objects exchanged and their characteristics and attributes, activities and functions related to the management of a plant, but does not specify about the implementations (tools) hosting these specific operations nor the precise assignment to one of the levels 2, 3 or 4. Realisations depend on individual customer needs and the tool manufacturer's strategies. For instance, maintenance management operation may typically be assigned to a Computerised Maintenance Management System (CMMS), a manufacturing execution system—both being typical Level 3 tools—but also to an Enterprise Resource Planning (ERP), dedicated to Level 4, or a Distributed Control System (DCS) that can be found at Level 2. Borders between these systems become floating.

Individual operations can be assigned to different specific manufacturing operations management areas—production operations management, quality operations management, maintenance operations management or inventory operations management. Having a look into these areas, individual activities (like resource management, detailed scheduling, dispatching, tracking, analysis, definition management, data collection, execution management [21]) can be identified to be executed within single or distributed sources. These functions can be implemented using different technologies. Currently, there is no standardisation regarding technologies to be used for implementing these functions.

2.2 Data Flow Within Automation Systems

The ways of communicating between the levels are different. Levels 1 and 2 are commonly connected through either point-to-point cabled solutions (4–20 mA current loop) or through fieldbuses (Modbus, Profibus, etc.). Ethernet and serial connections

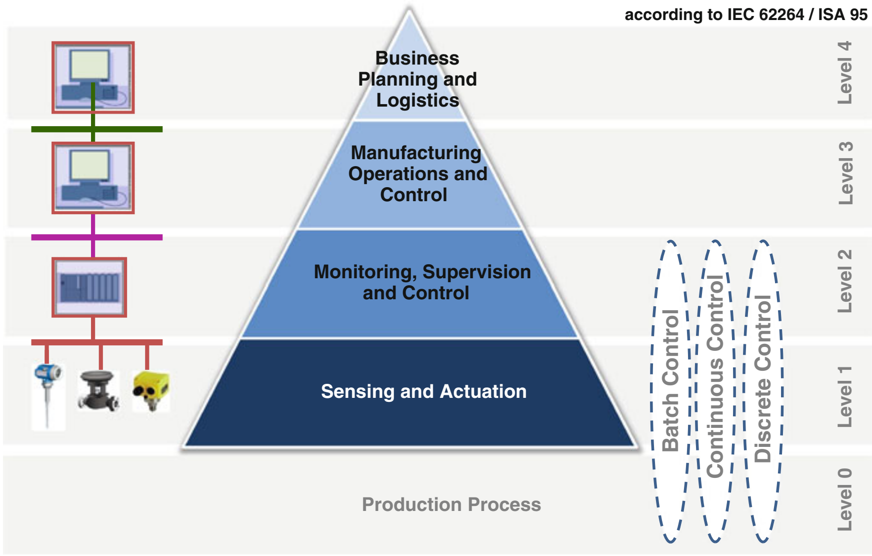

Fig. 2.1 Functional hierarchy according to (IEC 62264-3) [21, 39]

are used to an increasing extent as well. Fieldbuses and Ethernet can give an impression of a standard solution but the data exchange protocol on top of them is often proprietary, which leads to vendor lock-in. Some vendors start with a standard (electrical) interface but use a different non-standard connector, another kind of vendor lock-in. Because of this, end-users often must buy adapters, e.g. a converter to connect the serial port on the device to a port on the control system.

Figure 2.2 highlights some of the diversity of interfaces between the different levels and tools, which may even be distributed across the life cycle of a production system [25]. Profibus, Modbus or Foundation Fieldbus can give an impression of a standard solution. Fieldbuses standardise how to communicate; for instance, in order to configure a Profibus master to communicate with a slave, configuration files called GSD are required. These files specify the supported transmission speed and size of supported data buffers. GSD files can also hint about the interpretation of data. Additionally, semantics of data may be defined within device profiles, as done for Profibus PA or Foundation Fieldbus [13].

Monitoring of processes and automation equipment is an inherent pre-condition for keeping the production process alive and hopefully at near-optimal conditions to fulfil the business goals in the short, medium and long terms. It has to be guaranteed that data are provided:

- to the right application,
- in the quality (right semantics and syntax) needed for the consuming application,
- in right time (real-time) and sequence.

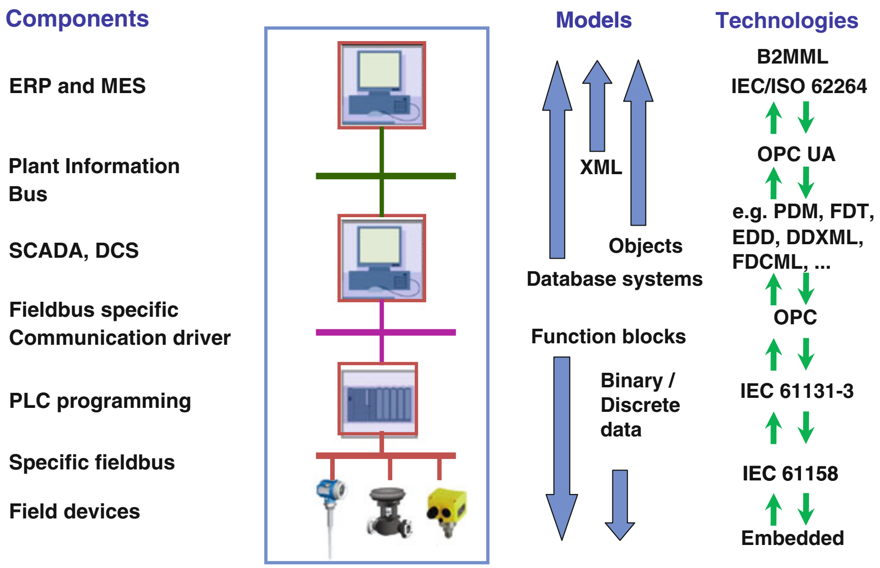

Fig. 2.2 Diversity of data and interfaces

Different applications raise specific requirements about the provision of data. Specifically for closed-loop control, data today must be retrieved in a cyclic manner. These sample times must be in a range that is suitable to the time-constraints of the controlled process. For that purpose, within a DCS, data are either polled internally from the DCS IO-cards, e.g. while accessing field devices through drilled lines supporting standardised signals (e.g. 4–20 mA analogue signal), or retrieved from remote-IO components via digital communication, following appropriate sample times, as described above.

Accessing process values within field devices through fieldbus communication is mainly done in a polling-based manner, e.g. Profibus with token-passing bus access, or based on the Publisher-Subscriber principle following configured cycle times (as done for Foundation Fieldbus). Transmitting data through digital protocols allows the association of status information (process and/or device related) to the process value. For instance, with Profibus PA communication, analogue process values are typically each transmitted as a Floating Point value associated with an 8-bit status in a single data structure each time a value is transmitted.

Considering the example of Profibus PA, the status Byte contains general information about device status, limit crossing of the process values measured, the validity of the process value as well as information to indicate maintenance demand. In cases when a failure is indicated, additional detailed information can be retrieved from the field device by individual a-cyclic requests. This construction of data allows interpretation by different types of applications:

- The process value, validity and limits are useful for the control application itself.
- This information will also be useful for supervision applications.
- Device status information is specifically needed for maintenance applications (Plant Asset Management).
- Production management applications will operate on more condensed data, representative of the production output. Such information typically is built by PLC or DCS based on information described above.

2.2.1 Use of Data for Supervision

SCADA deals with the gathering of data in real-time from remote locations in order to control and monitor the process, including data aggregation and presentation to the user. SCADA is commonly used in a broad range of application fields, like power plants as well as in oil and gas refining, telecommunications, transportation, and water and waste control, to mention a few. A typical SCADA system, as roughly depicted in Fig. 2.3, consists of several subsystems [23, 26] notably:

- A Human–Machine Interface (HMI) where the information is depicted and is used by human operators to monitor and control the SCADA linked processes.
- A computer which does the monitoring (gathering of data) as well as control (actuation) of the linked processes.
- Remote Terminal Units (RTUs) that are collecting data from the field (deployed sensors make the necessary adjustments and transmit the data to the monitoring and control system).
- Programmable Logic Controllers (PLCs) that are used as an alternative to RTUs since they have several advantages (like ability to deploy and run control logic) over the special-purpose RTUs.
- A communication infrastructure connecting all components.

SCADA systems include hardware and software components. The hardware gathers and feeds data into a computer that has a SCADA software installed. The software in a computer then processes these data and presents it in a timely manner. SCADA also records and logs all events into a file or sends it to a user terminal. These user terminals come in the form of Human–Machine Interface (HMI) or User Interface (UI) displays that allow the system to show data and warn when conditions become hazardous by generating alarms. Lastly, SCADA systems must ensure data integrity and appropriate update rates. Development of SCADA standards by industrial user groups and international standardisation bodies has allowed increased 'interoperability' of devices and components within SCADA systems [14]. Open protocols allow equipment from multiple vendors to communicate with the SCADA host. Many standards and specialised protocols exist with specific features.

Standards defining programming methods like IEC 61131-3 allow systems engineers to reuse code for logic operations and move easily between configuration interfaces. At the SCADA host level, the Open Connectivity via Open Standards

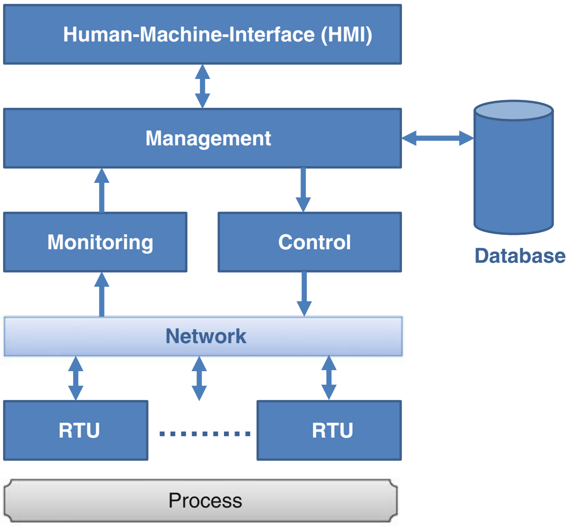

Fig. 2.3 Typical software architecture for a SCADA system

(OPC, previously OLE for Process Control) series of standards specifications have been widely accepted. Originally based on Microsoft's OLE Component Object Model (COM) and Distributed Component Object Model (DCOM) technologies, the specification defines a standard set of objects, interfaces and methods for use in process control and manufacturing automation applications to facilitate interoperability.

The OPC Foundation comprises a large group of vendor representatives dedicated to ensuring interoperability in industrial automation systems. The latest generation of SCADA system hosts the use of these OPC standards to provide advanced connectivity to user clients. The latest developments in OPC Foundation (www.opcfoundation.org/UA) denote: 'the new OPC Unified Architecture (OPC-UA) that is the next generation OPC standard (IEC 62541) that provides a cohesive, secure and reliable cross-platform framework for access to real time and historical data and events'.

These standards allow communications not only over serial links for dedicated communication channels, but also transfer of SCADA data over Ethernet with a TCP/IP protocol stack for Wide Area Networks (WANs) or Local Area Networks (LAN). Therefore, it is understood to benefit from an advanced high-speed, peer-to-peer communication service as well as improved device interoperability for process monitoring and automation, without the need for high cost of integration.

2.2.2 Use of Data Within Process Control Architectures

After decades of analogue single-loop controls, the early minicomputers started the transition to digital control systems in the 1960s. The Distributed Control System

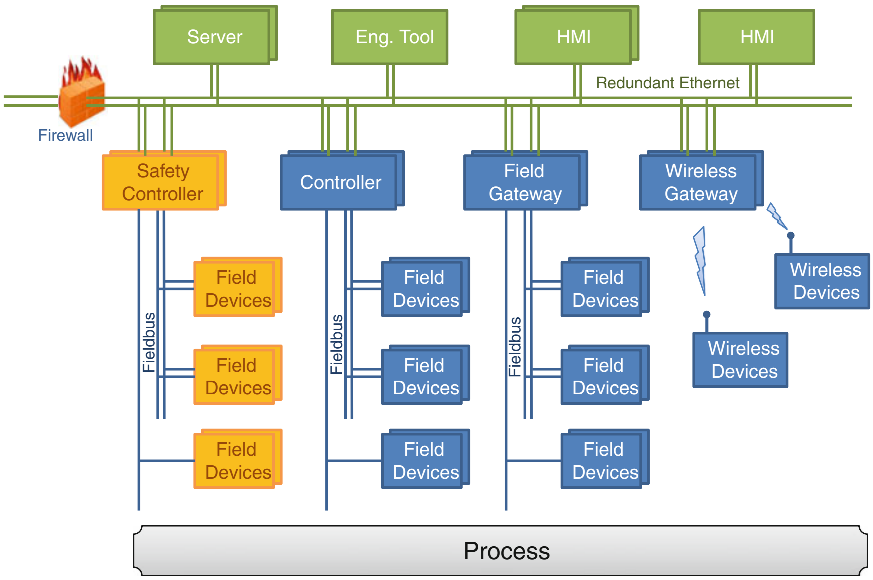

Fig. 2.4 State-of-the-art distributed control system

(DCS) was introduced at roughly the same time (1975) by Honeywell (TDC 2000) and Yokogawa (CENTUM). This was partly due to increased availability of micro-processors. The early DCSs were designed using proprietary hardware and software. The latest DCSs contain lots of Commercial off-the Shelf (COTS) components and IT standards are utilised whenever possible.

Today's state-of-the-art DCS has several nodes for different purposes as depicted in Fig. 2.4. The nodes are able to communicate using high-speed networks. Some of the nodes and networks are redundant and can tolerate single failure. The level of redundancy depends on industrial requirements, e.g. in the food and beverage industry the level of redundancy is quite limited while in the petrochemical industry almost all components are redundant. The DCS architecture is able to support a free combination of redundant and non-redundant components. It is also a very scalable architecture supporting all kinds of systems from very small (PC and some I/O channels) to very large and distributed systems (consisting of tens of thousands of I/O points and thousands of control loops). One of the goals in these systems is to secure the deterministic behaviour of the system at all levels in all circumstances.

The highest level nodes are 'Server', 'Engineering Tools' and human–machine interface 'HMI'. Today these are almost always PCs with Microsoft Windows operating system. The 'Server' contains all the configurations that are needed in the other nodes at runtime or in cold-start situations. It typically also contains data history collections, master alarm lists and perhaps interfaces to some other systems. These systems can be other DCS systems, Programmable Logic Controllers (PLC), Manufacturing Execution Systems (MES), Process Information Management Systems

(PIMS), Laboratory Information Management Systems (LIMS), Enterprise Resource Planning systems (ERP), etc.

These interfaces are usually implemented using OPC protocol. The ‘Engineering Tools’ node contains engineering tools for system structure definitions, controller applications, network definitions, HMI displays, etc. The ‘HMI’ node contains the graphical user interface which provides visibility to the process for the operator who is responsible for the process (or sub-process). The ‘Server’ is typically redundant and there are several HMI nodes to support several operators (but also to support HMI redundancy). The Engineering Tools node does not need to be redundant since it is not required in normal operations.

The highest level network is the ‘Redundant Ethernet’ network which takes care of the communication between controllers, gateways, servers, engineering tools and HMI. It is typically redundant and the swap between active and passive network is transparent to applications in case of hardware/network failure. Both networks use independent network switches and these are isolated from other networks by firewalls. The protocol stacks typically support low-level TCP/IP and UDP/IP communication but the deterministic behaviour is guaranteed with proprietary protocols that take care of the network utilisation.

The ‘Controller’ node is an important node in the system. It is where the most important control algorithms (closed and open loop) and logic are running. These nodes use proprietary hardware and software environments. The hardware supports some kind of non-volatile memory and high-speed redundancy. In many cases it is also designed to survive in harsh environments. The execution environment runs on a hard real-time operating system executing typically function block configuration but also other programming languages (in a time-constrained manner).

The controller is either connected directly or through a ‘Field Gateway’ to the fieldbuses. The fieldbuses are based on (mostly de facto) standards. The most popular fieldbuses are Foundation Fieldbus (H1 and HSE), PROFIBUS (DP and PA) and Ethernet-based PROFINET. The fieldbuses and field devices can be redundant or non-redundant. The protocols used in these fieldbuses can guarantee the deterministic behaviour when delivering critical data. The less time-critical data (e.g. diagnostics data) is transferred in the remaining time slots. It is also possible to add digital communication to field devices that are connected using traditional analogue 4–20 mA cables using the HART protocol. It is also possible to integrate wireless devices into the DCS architecture using (redundant) ‘Wireless Gateway’. With these devices it is more difficult to guarantee the deterministic behaviour because of the less robust media. Several protocols are available, including WirelessHART, which maintains compatibility with existing HART devices, commands and tools.

In some industries, special industrial safety systems are required to protect humans, plants and the environment in case the process goes beyond the control limits. These are also part of the DCS architecture. The ‘Safety Controller’ contains special redundant hardware which is Safety Integrity Level (SIL) certified.

The controllers are able to transfer data to each other (peer-to-peer communication). These data are typically transferred cyclically with defined time intervals but can be also event based. The communication protocols at controller level guarantee the deterministic behaviour and in many cases data subscriptions are used. The alarms are always event based. The controllers (and other nodes in the system) generate alarms for the operator and these typically require human acknowledgement. The alarm list is maintained by the server and shown on the HMI nodes. The data for the HMI displays (graphical view of the process) show the live data that is transferred from the controllers. Usually, the data are only transferred to displays that are currently switched on.

The software architecture inside the distributed control system is still based on object-oriented principles. Services are available but in many cases they are not created as granular components. Also, the interfaces are typically used for direct (local) method calls or direct data access rather than standards-based open remote interfaces. Online service discovery is also limited. Moving to SOA in distributed control systems would clearly bring architectural benefits and ultimately benefits for the users through services being more open, easy to find and accessible for external applications. It would also simplify the development and maintenance of the distributed control system and support new capabilities.

2.2.3 Use of Data for Production Management

Enterprises are moving towards service-oriented infrastructures that bring us one step closer to the vision of 'real-time enterprises' [27]. Applications and business processes are modelled on top of and using an institution-wide or even cross-institutional service landscape. For any solution to be easily integrated in this environment, it must feature a service-based approach.

One can realise a 'real-time enterprise' via strong coupling of the enterprise concepts domain and the device-level service domain. Nowadays, there is multi-step cooperation between the two layers, which in practice translates into the coupling of Enterprise Resource Planning (ERP) with Manufacturing Execution System (MES) and Sistributed Control System (DCS). By integrating device-level services with higher level enterprise services, timely information can flow to business processes and enhance existing applications.

As the whole enterprise is seen as a complex ecosystem, every process may affect several others in the system and therefore need to be managed in an integrated way. This includes:

- Warehouse and production management—Management of inventory across multiple warehouses, tracking of stock movements and management of production orders based on material requirements planning.
- Customer relationship management.
- Purchasing—Automation of procurement process from purchase order to vendor invoice payment.
- Reporting—Real-time information with detailed reports.

There are several IT systems that exist in the factory or plant floor today and data that are collected at various levels. At the lowest level is SCADA systems as repositories of field real-time massive data as they collect data from the PLCs and sensors that are connected to the machinery on the factory or plant floor. At the next higher level are MES that track all customer orders, schedules, labour, resources and inventory across the production line by shift. At the uppermost ERP and other enterprise solutions like Supply Chain Management (SCM), etc., plan and record transaction data to measure variance against set performance targets, etc.

Unfortunately, in many manufacturing companies today, these three layers are still not fully integrated. As a consequence, companies often employ large numbers of people to punch in or import redundant production batch data from their MES to their ERP systems. This is not only a wasteful and costly exercise but also introduces human errors in the data entry process. Even if done in an automatic way, this usually includes huge delays (sometimes in days), which prohibits the managers from getting a real-time/right-time picture of factory performance, variance from set targets as well as order/materials/machine/labour/quality/maintenance exceptions and issues that may arise in the factory. The latter may be translated into lost opportunities, e.g. failure to optimise production or even unhappy customers due to delayed shipments.

While the SCADA and MES layers tend to be integrated at most companies, it is equally likely that the heterogeneity of this environment comprising home-grown, legacy and point applications from multiple vendors with differing architecture platforms may result in disconnections in this layer as well. This tends to further exacerbate the problem.

The business implication of any exception or the ability to compare actual manufacturing performance against set targets is not evident until MES data and exceptions from the factory floor hit the ERP system. ERP in essence, if integrated seamlessly with the factory MES layer, provides the business context for manufacturing transactions, exceptions and issues captured on the factory floor. The bottom-line implication for manufacturers is that the disconnect between the Shop Floor (Factory MES) and the Enterprise Top Floor (ERP) costs them millions of Euro through waste, reject, re-orders, expedites, preventable material/machine/labour/quality issues that are detected too late, for enterprises to proactively resolve them.

Based on these considerations one can identify distinct directions towards the organisational structure of a production site and the topological or architectural characteristics. From the organisational point of view, the business is typically structured in a similar way to the levels and operations defined by IEC 62264, however, it might be better to express this in the opposite manner, i.e. that the standard is following what has been developed over the past years. Structures, skills, responsibilities, professions, education, etc., have been established focussing these organisational matters. It is questionable if, and how fast this may change in the future.

2.3 Integration Technologies Between Layers and Applications

Today, integration of Legacy Systems into new state-of-the-art systems has becoming an elementary task for each solution provider or engineering company. Legacy systems undergo continuous changes and modifications due to even more frequently changing requirements imposed by market needs. Normally, this progressively causes a significant increase in the complexity of existing systems [7]. The main problem with the integration process is the heterogeneity among systems. The heterogeneity issue [22] can be divided into:

- Technological Heterogeneity, e.g. different hardware, operating systems, communication protocols for accessing data and programming languages.
- Semantic Heterogeneity (e.g. the same names of data sources but different meaning or different names associated with the same meaning).

From the software architecture point of view, in order to integrate legacy systems, the role of each subsystem or component that is to be integrated has to be defined along with the interfaces and building object wrappers for each subsystem. An integration approach, where the system developer is required for knowing the internals of the legacy system is known as White-Box approach and an integration approach that only requires knowledge of the external interfaces of the legacy system is known as Black-Box approach [7, 11]. In order to integrate legacy devices into state-of-the-art automation systems, legacy adapters can be used, being composed of [31]:

- State-of-the-Art Interface Layer (required to communicate with the state-of-the-art system, configuration capabilities have to be provided),
- Integration Layer (used for protocol transformation, data and semantics transformation; configuration capabilities have to be provided),
- Legacy Systems Interface Layer (provides the communication capabilities for exchanging data with legacy components, configuration capabilities have to be provided).

There are different ways to integrate legacy systems using adapters, e.g. by utilising gateways or mediators. Besides these general concepts, specific technologies and concepts for integration of data are used or approached in today's automation systems, e.g. Electronic Device Description (EDD), Field Device Tool (FDT), Field Device Integration (FDI), OPC Unified Architecture (OPC-UA).

2.3.1 Integration Using Gateways and Mediators

Using gateways is a well-proven concept for integrating/connecting devices, attached to different networks. It is used to transform protocols as well as the syntax of data. Semantic integration is harder to achieve. Nevertheless, it is possible to do transformation between data centric approaches, as typically followed by fieldbus concepts and service-oriented approaches.

A gateway, as defined in the FP6 SOCRADES [25] and FP7 IMC-AESOP projects, is understood to be a device that controls a set of lower level non-service-enabled devices, each of which is exposed by the gateway as a service-enabled device (as depicted in Fig. 2.5). This approach allows the gradual replacement of limited-resource devices or legacy devices by natively service-enabled devices without

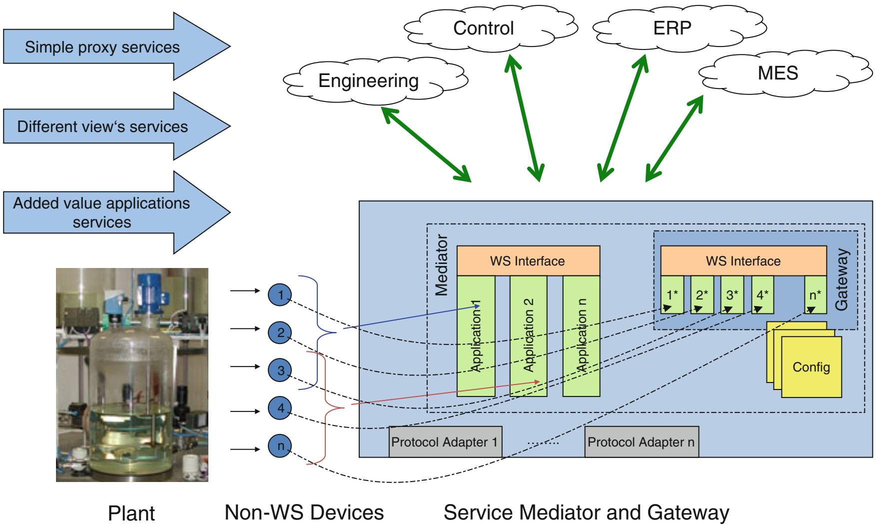

Fig. 2.5 Gateway and mediator concepts for integration of devices [1]

impacting the applications using these devices. This approach is used when each of the controlled devices needs to be known and addressed individually by higher level services or applications.

The mediator concept is based on the elaboration of the gateway concept, while adding additional functionality to the gateway. Originally meant to aggregate various data sources (e.g. databases, log files, etc.), mediator components have evolved with the advent of Enterprise Service Bus (ESB) [17]. Service mediators are now used to aggregate various services in SOA. As such, a mediator can be seen as a gateway, except that it can hide (or surrogates) many devices, not just one. However, service mediators also go beyond gateways since they introduce semantics in the composition. Mediators aggregate, manage and eventually represent services based on some semantics, e.g. using ontologies.

2.3.2 *Electronic Device Description*

An Electronic Device Description (EDD) is based [20] on a formal language called Electronic Device Description Language (EDDL). This language is used to describe completely and unambiguously, what a field instrument looks like when it is seen through the 'window' of its digital communication link. EDD includes descriptions of accessible variables, the instrument's communication related command set and operating procedures such as calibration. It also includes a description of a GUI structure which a host application can use for a human operator. The EDD, written in a readable text format, consists of a list of items ('objects') with a description of the features ('attributes' or 'properties') of each.

The major benefit of EDD for device suppliers is that it decouples the development of host applications and field devices. Each designer can complete product development with the assurance that the new product will interoperate correctly with current and older devices, as well as with future devices not yet invented. In addition, a simulation program can be used to test the user interface of the EDD, allowing iterative evaluation and improvement, even before the device is built.

For the user, the major benefit is the ability to mix products from different suppliers, with the confidence that each can be used to its full capacity. Easy field upgrades allow host devices to accept new field devices. Innovation in new field devices is encouraged. The EDD is restricted to the description of a single device and use in a mostly stand-alone tool, preferably for commissioning the field devices. Due to the nature of EDD such tools are based on interpreter components suitable to the EDDL.

Software tools for automation are complex, and implement a lot of know-how. The number of sold products is relatively low in comparison with office applications. The definition of standardised device description languages increases the potential users of such tools and also encourages the use of fieldbus-based automation.

2.3.3 Field Device Tool

In order to maintain the continuity and operational reliability of process control technology, it is necessary to fully integrate field devices as a subcomponent of process automation [36]. To resolve the situation, the German Electrical and Electronic Manufacturers' Association (ZVEI) initiated a working group in 1998 to define a vendor-independent Field Device Tool (FDT) architecture, the specification of which is maintained and refined inside the FTD Group (www.fdtgroup.org).

This FDT concept defines interfaces between device-specific software components (DTM—Device Type Manager) supplied by device manufacturers, and engineering systems supplied by control system manufacturers. The device manufacturers are responsible for the functionality and quality of the DTMs, which are integrated into engineering systems via the FDT interface. With DTMs integrated into engineering systems, a unified way of creating the connection between engineering systems (e.g., for PLC applications) and currently inconsistent field devices becomes available. The FDT specification defines what the interfaces are. DTMs act as bridges between the frame-application and field devices. Several technical documents on FDT summarise the available features (more info available at www.fdtgroup.org/technical-documents).

2.3.4 Field Device Integration

Looking into the market situation, it can be noticed that both aforementioned technologies for device integration, i.e. EDD and FDT, are competing on the market [16]. On one hand, benefits of EDDL such as robustness, independence from the operating system and backward compatibility are promising characteristics for the system

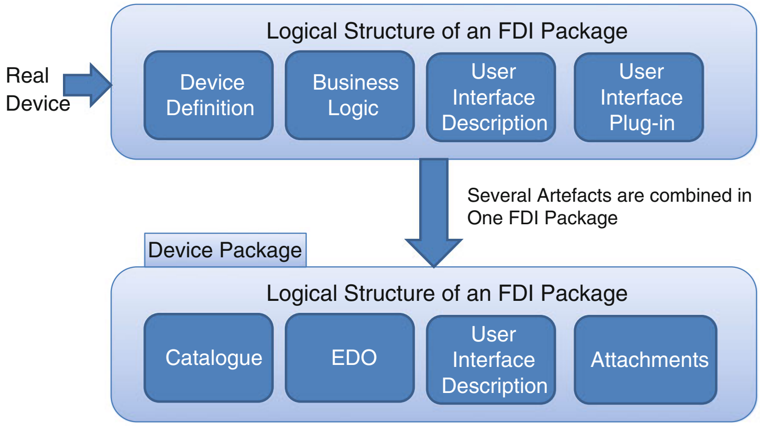

Fig. 2.6 Structure of an FDI device package [16]

integrator or the end-user. On the other hand, the FDT approach provides potential to allow the device vendors to represent their brand label, realising highly sophisticated user interfaces to the end-user. FDT components may be easily plugged into a DCS or other commissioning and operations management tools, which is seen by the user as a useful service.

The system providers have to handle more and more complex systems. Such systems will be less homogeneous and more distributed, having different network technologies, including gateways between them or requiring worldwide online access. Although existing solutions may offer such features they will often be proprietary. EDDL and FDT are the basis of Field Device Integration (FDI) [16], which is targeting to provide a way of migration of both technologies (EDDL and FDT). It is intended to take advantage from the more promising concepts of both technologies.

In FDI the device is represented by an FDI device package, Fig. 2.6, and covers all information needed for the integration of the field device into the automation system. The device vendor provides the FDI device package. It replaces the EDD or DTM and consists of several components as shown in Fig. 2.6, but the end-user now has to install only one file—the FDI device package—in the system. Thus, this is a significant improvement in handling such a complex information pool.

The FDI device package consists of logical blocks such as device definition, business logic, user interface description and user interface plug-in [16]. Device definition describes the parameters of the device and its internal structure, e.g. blocks or modules. Business Logic ensures the consistency of the device parameters (this means also the consistency of the device model, see above). Examples of such consistency rules are dynamic conditions or relations between parameters. Thus, parameter values could be changed depending on the device status/device configuration. GUI elements could be available as descriptive elements (user interface descriptions) or as programmed components (user interface plug-ins).

2.3.5 OPC: Unified Architecture

Classical OPC is a technology widely used as a basic communication platform for integrating data for supervision and control purpose based on information models defined. Many products (such as PLC, DCS and SCADA devices) exist on the market supporting OPC server or client components. During the last years the original OPC specifications, based on Microsoft COM/DCOM, were replaced by new interoperability standards, such as Web services. Consequently, the OPC Foundation published the OPC Unified Architecture (OPC-UA) [32].

The transition towards this unified architecture started with the development of the OPC XML DA specification, which introduces the use of XML, thus allowing the flow of information beyond corporate firewalls and permitting cross-platform connectivity via Simple Object Access Protocol (SOAP) and Web services through the Internet [19]. The limitations of OPC-UA however, are mainly evident at the factory level, namely at the device level. While OPC-UA allows the integration of process control devices with SCADA and even MES systems, the information offered by low-level devices can only be accessed through process control systems. In order to further expand the reach and flow of information, device integration standards such as Field Device Tool (FDT) and Electronic Device Description Language (EDDL) can be used [19].

Several technology supporting organisations—such as PROFIBUS International (PI), Fieldbus Foundation (FF), HART Communication Foundation (HCF), or others—started investigating the potential use of OPC-UA to take advantage of this basic technology. As an example, PLCopen and OPC Foundation are undertaking common activities to jointly define a common information model. Information models have been developed for Electronic Device Description (EDD) and now also for IEC 61131 PLC. This development ensures that field devices that are described in EDD and in future that are represented by PLC proxies can be accessed by OPC-UA Web services (more info is available at www.plcopen.org).

OPC-UA uses client-server architecture with clearly assigned roles. Servers are applications that expose information following the OPC-UA information model, where each server defines an address space containing nodes of the OPC-UA model. These nodes represent real physical or software objects. Clients are applications retrieving information from servers by browsing and querying the information model. Both types of applications can be developed using an API that isolates the application from the communication stack. Figure 2.7 gives an overview of the flexibility and extensibility of the OPC-UA architecture.

Interoperability and adaptability of the standard are reachable through several complementary features of OPC-UA:

- Extensible object model;
- Rich set of services;
- Scalability;
- Reliability, Redundancy and Performance;
- Security;

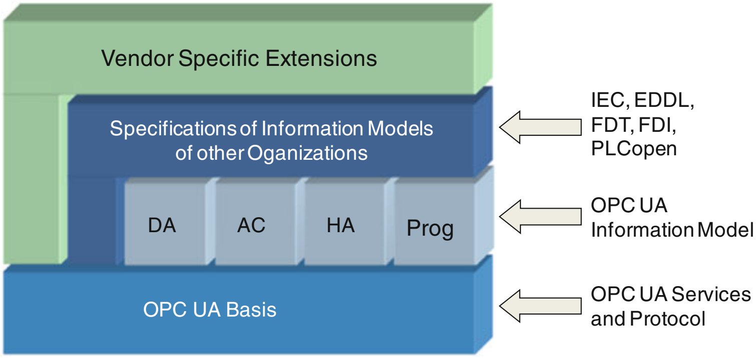

Fig. 2.7 Overall OPC-UA architecture [32]

- Backwards compatibility;
- Standardisation at the protocol level;
- Isolation of the application from the communication stack through the client or server API.

2.4 Engineering of Production Systems

There is an on-going trend towards higher levels of automation in process control systems [15, 31] with increasing levels of autonomy in control and monitoring. Today's automation/business systems are moving to a 'Smart' environment such as smart devices, smart systems, smart organisations and smart cities, where Smart may be defined as systems that exhibit (i) extended functionality, (ii) multi- functionality, (iii) self-diagnosis, (iv) configurability and (v) connectivity.

With the increased use of Commercial off-the Shelf (COTS) technologies, the network infrastructure of the DCS and network architecture for plant information become increasingly interdependent. The prevalence of Ethernet at every level of an organisation, especially in green-field sites provides shop-floor systems with the infrastructure for data acquisition, analysis and integration with other enterprise systems [37]. This also creates problems with the proliferation of data, which requires integration and management.

Tools and methods are required to manage this and make complex time-dependent data integrated from disparate sources available to other systems within the enterprise in a consistent manner. Users of these systems are becoming more demanding too, it is expected that timely data should be available 'anywhere, anytime and on multiple platforms' (e.g. mobile and web devices). Additionally, users will expect systems to be richer not only in content but graphically too, and they will expect more interactive graphical systems with emphasis placed on design of the user interface as well as the functionality being offered [29].

The current trend in manufacturing system design tool development consists in merging system mechanical and control design software in a single environment in order to break the communication barrier that commonly exists between mechanical and control engineers and which translates into difficulties to coordinate two complex, but separated design processes. This approach is dominated by Siemens (Process Simulate) and Dassault Systems (Delmia) providing solutions that can potentially take CAD models and provide 3D kinematic simulations to validate the mechanical design and engineering process, and generate code for deployment on PLCs [18]. A related approach that has gained popularity uses Winmod for modelling control behaviour and Invision for 3D modelling, which allows the virtual commissioning and simulation of automation systems [33].

Traditional shop-floor applications are likely to be superseded by cloud-based applications (where hardware control is not an issue), and with the introduction of software-as-a-service (‘SaaS’) models, it means that software will be less hardware dependent and more dynamic in nature as service upgrades should happen without shop-floor intervention [6, 28].

Smart network attached devices are becoming more and more powerful and cheap to produce; the expected resultant explosion in these devices will lead to more widespread use of DCS, where devices will cooperate in a peer–peer way to meet the system goals [8]. These devices will drive engineering tools and methods to handle the building and development of systems as a set of cooperating modules or components whose application logic is either centralised and the device behaviour is orchestrated, or the application logic is distributed to the devices and the overall behaviour is choreographed. In either method, tools are capable of integrating devices from different vendors and domains (e.g. business, external and automation components). One promising methodology for achieving this is the use of AutomationML (www.automationML.org), which is described as a neutral data format for automation engineering.

In addition to these design and development tools, engineering tools are required to support the complete life cycle of an automation system. In many cases, these virtual engineering approaches are used to create automation systems and provide visualisations that can be used as a catalyst for communication and understanding between disciplines (such as mechanical, control and safety engineering) and even the supply chain, but once the system is commissioned these models are not kept up-to-date with changes that occur during its life, due to the time and cost associated with maintaining the original models.

There is a requirement for lightweight visualisations that may be used to aid in diagnostics and maintenance; these tools should be directly linked to the automation systems such that changes may be quickly and simply made in the engineering tools validated and then deployed directly on the system, or when changes are made directly on the physical system when the model will reflect these changes implicitly. In this way greater return on investment in modelling and simulation can be achieved.

If this trend of building heterogeneous systems continues, systems will become more modular and componentised. This should enable systems to be built from a blend of the best custom-built apps and off-the shelf-components, which could make the market more open and more competitive. The introduction of these will be dependent on the ability of such systems to be maintained effectively and to ensure that the production downtime is still kept to a minimum. Acceptance of such technologies is likely to depend on familiarity of control representation (e.g. ladder, timing/Gantt chart, function block diagrams), such that engineers will be able to understand and maintain them using their core knowledge.

Advances in active tagging result in direct or indirect tagging of devices, work pieces, employees, etc., and as they become cheaper and more widely used, future automation systems should be capable of using this information and integrating it with control to enhance performance (e.g. live inventory control), safety (e.g. employee tracking) and maintenance (e.g. location of mechatronic devices). In conclusion, technological and infrastructural advances in automation system design manufacture and deployment is happening rapidly, however, engineering tools capable of effectively supporting and exploit these advances are severely lacking or fragmented. The challenge is therefore to provide engineering tools and effective interoperability between such tools for the next generation of DCSs.

2.5 Towards SOA-Based Automation

Among the biggest challenges faced by manufacturing enterprises are the constant demands to change their processes and products and still be able to manage the inherent complexity in all levels of their production environment. In order to provide the IT support needed to cope with these challenges, appropriate ways of designing automation software systems are required. As a consequence, factory automation providers are integrating the SOA approach in their solutions for Manufacturing Execution Systems (MES), Enterprise Resource Planning (ERP) or Enterprise Asset Management (EAM) systems.

However, many challenges remain when applying the service technology to the shop floor devices characterised with limited resources and real-time requirements. At this level, the interactions are still carried out using different fieldbus and industrial Ethernet protocols with restricted interoperability across technology borders. This limits the ability to enforce plant-wide, seamless integration of processes and services leading to complex systems for monitoring and control that are heavily dependent on the interactions with various resource constrained shop floor devices such as sensors and actuators.

2.5.1 Building Service-Based Infrastructures

To overcome this situation and to address integration of very large numbers of subsystems and devices (including field level devices) within a harmonised networking architecture, several European collaborative projects such as IMC-AESOP [26], SIRENA [2], SODA [12], SOCRADES [9, 38], etc., investigated Web services at the device level and integrated these devices with MES and ERP systems at upper levels of an enterprise architecture [10, 24, 27]. The first results shown in pilot applications running in the car manufacturing, electromechanical assembly and continuous process scenarios have been successful [4], confirming that the use of cross-layer service-oriented architectures in the industrial automation domain is a promising approach. Additional examples, coming from the IMC-AESOP project are presented within Chaps. 7–10, highlighting the use of Web service technologies within the domain of control and monitoring of batch and continuous processes.

The FP6 SOCRADES project evaluated several SOA solutions, applicable at the device level in the context of manufacturing automation. The SOCRADES (DPWS based) solution was provided as a complete open-source software component, which was embedded in several devices and tools, and was demonstrated in electronic assembly demonstrators, continuous process control and in interoperability trials. A potential merger between DPWS and OPC-UA was also identified [3, 35]. Potential solutions were identified to reduce the costs of embedding DPWS in very simple devices. Generic and application Web services were identified, specified and implemented in prototype applications.

To overcome the often-poor integration between engineering methods and tools, IMC-AESOP looked at tools and methods established, or emerging, in the process control sector, plus applicable approaches from other domains relevant to an SOA-based engineering approach. The engineering requirements of large-scale process control systems were considered likely to be somewhat different from the smaller scale systems previously considered in SOCRADES, i.e. in terms of control and monitoring, traceability and integration with management systems, data acquisition and reporting, and system reliability and security [30].

The IMC-AESOP project considered the state of the art in engineering tool life cycle engineering capabilities and related user application requirements from the perspectives of:

- Monitoring;
- Control;
- Enterprise and management integration systems, e.g. application of SCADA and MES;
- SOA engineering methods, tool and the application of Web services;
- System visualisation, e.g. 2/3D system visualisation;
- Simulation methods, e.g. optimisation and key performance controls, prediction of system behaviours;
- Quality control;
- Environmental factors, e.g. energy optimisation.

Based on the findings it is considered that, in an SOA context, engineering applications of the future will need to:

- *Provide integration*. People and computers need to be integrated to work collectively at various stages of the product development and even the whole product life cycle, with rapid access to required knowledge and information. Heterogeneous sources of information must be integrated to support these needs and to enhance the decision capabilities of the system. Bi-directional communication environments are required to allow effective, quick communication between human and computers to facilitate their interaction.

- *Be heterogeneous*. To accommodate multi-vendor and multi-purpose software and hardware in both manufacturing and information environments.
- *Be interoperable*. Heterogeneous information and control environments may use different programming languages, represent data with different representation languages and models and operate in different computing platforms. Yet these subsystems and components should interoperate in an efficient manner.
- *Be open and dynamic*. It must be possible to dynamically integrate new subsystems (software, hardware or manufacturing devices) into or remove existing subsystems from the system without stopping and reinitialising the working environment.
- *Be agile*. Considerable attention must be given to reducing product cycle time to be able to respond faster to customer desires. Agile manufacturing is the ability to adapt quickly in a manufacturing environment of continuous and unanticipated change and thus is an essential component in manufacturing strategies for global competition. To achieve agility, manufacturing facilities must be able to rapidly reconfigure and interact with heterogeneous systems and partners.

The advantage of Service-Oriented Architectures (SOA) in the industrial automation domain are manifold including: device virtualisation using Web services; automatic composition, orchestration and configuration of distributed automation functions and systems by means of service-based applications; use of technologies at the research edge providing real-time and large-scale industrial automation and control applications. However, as identified by the SOCRADES project the significant benefits assume that several challenges will also be adequately addressed [38].

2.5.2 Virtualisation of Smart Embedded Automation Devices with Web services

Typical production equipment like transport units, robots, but also sensors, valves, etc., are considered as modules integrating mechanic, electronic, communication and information processing capabilities. This means that the functionalities of the modules are exposed via Web services into a network, as depicted in Fig. 1.2 [5]. Embedding Web service protocols into the automation device, e.g. DPWS or OPC-UA [34] allows the transformation of traditional industrial equipment into the nodes of an information-communication-network. Such nodes will be able to expose and also to consume ‘Services’. Moreover, depending on the position and inter-relation of such nodes to other nodes of the network, it becomes necessary to compose, orchestrate and/or choreograph services.

The virtualisation of a mechatronic module transforms it into a unit able to ‘collaborate’ with other units. That is, a module that communicates with others, exposing or consuming ‘Services’ related to automation and control functions. Recent trends

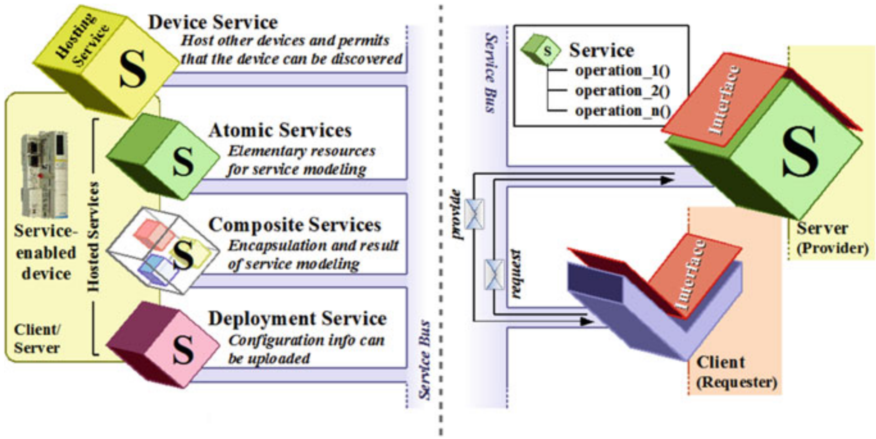

Fig. 2.8 Web service classification for SOA-compliant smart embedded device

in the technology developments associated to automation devices facilitate the virtualisation: Web service protocols are now embedded into a chip, integrated into industrial automation and control devices.

Different specifications of a collaborative mechatronics module and the corresponding smart automation device are virtualised and the resultant 'Services' can initially be classified according to the position and offered functionality of the smart device. Figure 2.8 shows an initial classification of the 'Web services' that will be exposed to the network and will immediately be ready to be consumed/requested from other nodes of the SOA-based network.

2.5.3 Configuring a Shop Floor as an SOA-Based Collaborative Automation Network

A shop floor composed of smart embedded devices that follow the specifications already discussed appears as a flat automation architecture, where each component has a Web service interface and may take part in various orchestrations collaborating with other service-enabled devices and systems.

Within Fig. 2.9 the block with the denomination 'Service Orchestration' represents a module that is able to compose and orchestrate 'Services'. This logic function will be implemented in a centralised or distributed manner, depending on the kind of virtualised system. This means, orchestration (or even choreography) engines will be deployed into one or more smart automation devices, i.e., another SW component and processing engine inside the smart device. Devices are 'motors', 'valves', 'conveyors', 'storages', 'HMI', 'drives' and generally any mechatronic components with CPU-capability and embedded Web service stack. PLC and robot controllers can also be transformed into 'service producer/consumer' integrating Web service capabilities.

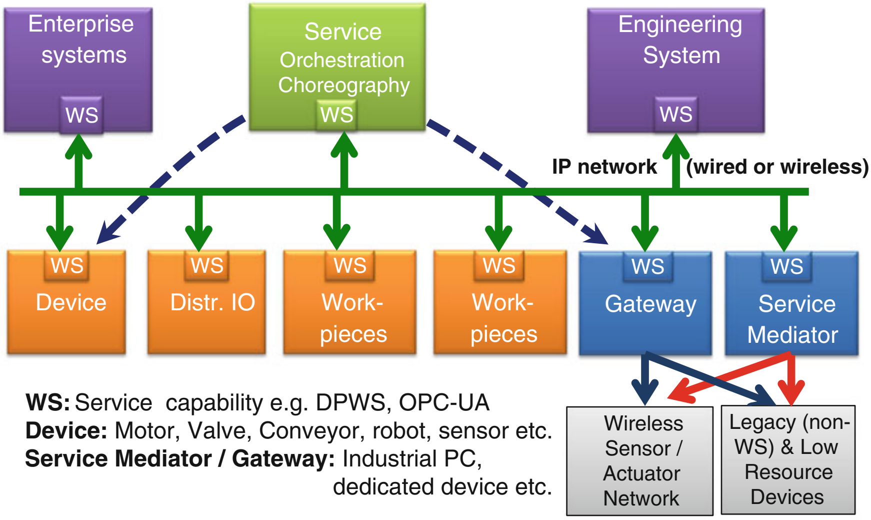

Fig. 2.9 Flat SOA-based technical architecture of production systems

One of the major outcomes of the Web service-based virtualisation of a shop floor is the possibility to manage the whole system behaviour by the interaction of Web services, i.e. exposition, consumption, orchestration, choreography, composition of the different kind of services exposed by the different SOA-compliant smart devices and systems.

A deeper analysis of the SOA-based automation systems shows that the SOA-based virtualisation, applied to an enterprise, makes a clear transformation (from the architectural point of view) of the traditional hierarchical ISA-95 compliant enterprise architecture into a 'logical' flat architecture [28]. This major and fundamental outcome of the Web service-based virtualisation of a shop floor relies on the fact that the 'Services', when they are exposed using the same Web service-based protocol, are directly consumed, composed and/or orchestrated in an independent way from the source (where these services are physically originated). A Web service exposed by the MES component (located in the ISA-95 Level 3) can immediately be composed with a Web service generated by a valve (located in the ISA-95 Level 1).

Topological and architectural characteristics are driven by user or application needs with respect to latest, proven or acceptable technological capabilities. IMC-AESOP proposes and follows the idea of establishing a service cloud fulfilling today's requirements for production management systems. The composition of the cloud is targeted towards the suitability of supporting IEC 62264 operations and activities. Thus, one may still keep the organisational aspects established in today's production systems, while migrating to a future SOA-based underlying architecture, exploiting the desirable capabilities inherent to SOA.

2.6 Conclusion

PLC, SCADA and DCS systems are the basis for monitoring and controlling industrial applications at lower levels within the plant hierarchy. Upper levels are dominated by MES and ERP systems. Information exchange at lower levels is characterised by a data-centric approach utilising industrial serial fieldbus systems or Ethernet-based communication supported by appropriate engineering concepts and tools. Diverse standardisation activities towards interoperability have been undertaken in the past, focussing individual device classes, programming concepts or communication capabilities of neighbouring levels. All these, as roughly introduced within this section, are widespread across industrial sectors.

The more complex, large and diverse applications become, limits are reached by existing technologies requesting improvements or even new technologies to be introduced. On the other hand, innovations may only be as large and introduced as fast, as the user is able and willing to adopt them. Consequently, every work towards challenging targets must start from the base-ground. This chapter was dedicated to give a brief, not raising any claim for completeness, overview of the state of the art in industrial automation as well as some progress actually monitored. Based on this, the following chapters will introduce the innovative results of the IMC-AESOP project.

Acknowledgments The authors thank the European Commission for their support, and the partners of the EU FP7 project IMC-AESOP (www.imc-aesop.eu) for fruitful discussions.

References

1. Bangemann Th, Suchold N, Colombo AW, Karnouskos S (2012) Die Integration Service orientierter Architekturen in der Automation. In: Automation 2012, Baden-Baden, 13–14 June 2012. VDI-Berichte 2171, S 333–336, VDI Verlag GmbH, Düsseldorf. ISBN 978-3-18-092171-6
2. Bohn H, Bobek A, Golatowski F (2006) Sirena—service infrastructure for real-time embedded networked devices: a service oriented framework for different domains. In: International conference on networking, international conference on systems and international conference on mobile communications and learning technologies. ICN/ICONS/MCL 2006, p 43. doi:10.1109/ICNICONSMCL.2006.196
3. Bony B, Harnischfeger M, Jammes F (2011) Convergence of OPC UA and DPWS with a cross-domain data model. In: 2011 9th IEEE international conference on industrial informatics (INDIN), pp 187–192. doi:10.1109/INDIN.2011.6034860
4. Boyd A, Noller D, Peters P, Salkeld D, Thomasma T, Gifford C, Pike S, Smith A (2008) SOA in manufacturing—guidebook. Technical report, IBM Corporation, MESA International and Capgemini. ftp://public.dhe.ibm.com/software/plm/pdif/MESA_SOAinManufacturingGuidebook.pdf
5. Candido G, Jammes F, Barata J, Colombo A (2009) Generic management services for DPWS-enabled devices. In: 35th annual conference of IEEE on industrial electronics, 2009 (IECON '09), pp 3931–3936. doi:10.1109/IECON.2009.5415339
6. Chowanetz M, Pfarr F, Winkelmann A (2013) A model of critical success factors for software-as-a-service adoption. In: 7th IFAC conference on manufacturing modelling, management, and control, St. Petersburg, Russia

7. Chowdhury MW, Iqbal MZ (2004) Integration of legacy systems in software architecture. In: Specification and verification of component-based systems (SAVCBS) workshop at ACM SIGSOFT 2004/FSE-12
8. Colombo A, Mendes J, Leitao P, Karnouskos S (2012) Service-oriented SCADA and MES supporting petri nets based orchestrated automation systems. In: IECON 2012—38th annual conference on IEEE industrial electronics society, pp 6144–6150. doi:10.1109/IECON.2012.6389076
9. Colombo AW, Karnouskos S (2009) Towards the factory of the future: a service-oriented cross-layer infrastructure. In: ICT shaping the world: a scientific view. European Telecommunications Standards Institute (ETSI), Wiley, New York, pp 65–81
10. Colombo AW, Karnouskos S, Mendes JM (2010) Factory of the future: a service-oriented system of modular, dynamic reconfigurable and collaborative systems. In: Benyoucef L, Grabot B (eds) Artificial intelligence techniques for networked manufacturing enterprises management. Springer, London. ISBN 978-1-84996-118-9
11. Comella-Dorda S, Wallnau K, Seacord R, Robert J (2000) A survey of black-box modernization approaches for information systems. In: Proceedings of the international conference on software maintenance (ICSM'00), IEEE computer society, Washington, DC, USA, p 173. http://dl.acm.org/citation.cfm?id=850948.853443
12. Deugd SD, Carroll R, Kelly K, Millett B, Ricker J (2006) Soda: service oriented device architecture. IEEE Pervasive Comput 5(3):94–96, c3. doi:10.1109/MPRV.2006.59, http://dx.doi.org/10.1109/MPRV.2006.59
13. Diedrich C, Bangemann T (2007) PROFIBUS PA instrumentation technology for the process industry. Oldenbourg Industrieverlag, GmbH. ISBN 13 978-3-8356-3125-0
14. Fitch J, Li H (2010) Challenges of SCADA protocol replacement and use of open communication standards. In: 10th IET international conference on developments in power system protection (DPSP 2010). Managing the change, pp 1–5. doi:10.1049/cp.2010.0220
15. Ganguly J, Vogel G (2006) Process analytical technology (PAT) and scalable automation for bioprocess control and monitoring—a case study. Pharm Eng 26(1):8
16. Großmann D, Braun M, Danzer B, Riedl M (2013) FDI field device integration. VDE, Verlag. ISBN 978-3-8007-3513-6
17. 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. http://www.ceur-ws.org/Vol-168/MEDIATE2005-paper5.pdf
18. Hoffmann P, Maksoud TM, Schumann R, Premier GC (2010) Virtual commissioning of manufacturing systems: a review and new approaches for simplification. In: 24th European conference on modelling and simulation (ECMS 2010), Kuala Lumpur
19. Huovinen M (2010) Large-scale monitoring applications in process industry. Ph.D. thesis. Tampere University of Technology, Tampere. http://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/6512/huovinen.pdf
20. IEC (2006) Function blocks (FB) for process control—part 3: electronic device description language (EDDL) (IEC 61804–3)
21. IEC (2007) Enterprise-control system integration—part 3: activity models of manufacturing operations management (IEC 62264–3)
22. Karasavvas KA, Baldock R, Burger A (2004) Bioinformatics integration and agent technology. J Biomed Inf 37(3):205–219. doi:10.1016/j.jbi.2004.04.003, http://dx.doi.org/10.1016/j.jbi.2004.04.003
23. 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
24. Karnouskos S, Baecker O, de Souza LMS, Spiess P (2007) Integration of SOA-ready networked embedded devices in enterprise systems via a cross-layered web service infrastructure. In: Proceedings of 12th IEEE international conference on emerging technologies and factory automation (ETFA 2007), Patras (Best Paper Award), pp 293–300. doi:10.1109/EFTA.2007.4416781, http://www.iestcfa.org/bestpaper/etfa07/2194.pdf

25. 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
26. 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
27. Karnouskos S, Savio D, Spiess P, Guinard D, Trifa V, Baecker O (2010) Real world service interaction with enterprise systems in dynamic manufacturing environments. In: Artificial intelligence techniques for networked manufacturing enterprises management. Springer, London
28. 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
29. Kasik DJ (2011) The third wave in computer graphics and interactive techniques. IEEE Comput Graphics Appl 31(4):89–93. doi:10.1109/MCG.2011.64
30. Kirkham T, Bepperling A, Colombo AW, McLeod S, Harrison R (2009) A service enabled approach to automation management. In: 13th IFAC symposium on information control problems in manufacturing (INCOM), Moscow, Russia
31. Lindemann L, Thron M, Bangemann T, Grosser O (2011) Integration of medical equipment into SOA—enabling technology for efficient workflow management. In: 2011 IEEE 16th conference on emerging technologies factory automation (ETFA), pp 1–8. doi:10.1109/ETFA.2011.6059233
32. Mahnke W, Leitner SH, Damm M (2009) OPC unified architecture. Springer, Heidelberg. ISBN 978-3-540-68899-0
33. Makris S, Michalos G, Chryssolouris G (2012) Virtual commissioning of an assembly cell with cooperating robots. Adv Decis Sci 2012:11. doi:10.1155/2012/428060, http://dx.doi.org/10.1155/2012/428060
34. Mendes J, Leitao P, Colombo A, Restivo F (2008) Service-oriented control architecture for reconfigurable production systems. In: 6th IEEE international conference on industrial informatics 2008 (INDIN 2008), pp 744–749. doi:10.1109/INDIN.2008.4618201
35. Minor J (2011) Bridging OPC-UA and DPWS for industrial SOA. Master's thesis. Tampere University of Technology, Tampere. http://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/20954/minor.pdf
36. PROFIBUS (2001) FDT interface specification: specification for PROFIBUS device. Technical report, PROFIBUS User Organization. http://www.profibus.com/download/device-integration/
37. Sommer J, Gunreben S, Feller F, Kohn M, Mifdaoui A, Sass D, Scharf J (2010) Ethernet—a survey on its fields of application. IEEE Commun Surv Tutorials 12(2):263–284. doi:10.1109/SURV.2010.021110.00086
38. Taisch M, Colombo AW, Karnouskos S, Cannata A (2009) SOCRADES roadmap: the future of SOA-based factory automation
39. ZVEI (2011) Manufacturing execution systems (MES)—industry specific requirements and solutions. German Electrical and Electronic Manufactures' Association (ZVEI). http://www.zvei.org/en/association/publications/Pages/Manufacturing-Execution-Systems-MES.aspx. ISBN 978-3-939265-23-8

http://www.springer.com/978-3-319-05623-4

Industrial Cloud-Based Cyber-Physical Systems
The IMC-AESOP Approach
Colombo, A.; Bangemann, Th.; Karnouskos, S.; Delsing, J.; Stluka, P.; Harrison, R.; Jammes, F.; Lastra, J.L. (Eds.)
2014, XX, 245 p. 114 illus., 89 illus. in color., Hardcover
ISBN: 978-3-319-05623-4