Integration of Automation Devices in Web Service supporting Systems

Thomas Bangemann*, Christian Diedrich*, Matthias Riedl*, Daniel Wuwer*
Robert Harrison**, Radmehr P. Monfared**

*Institut f. Automation und Kommunikation, Werner-Heisenberg-Str. 1
39106 Magdeburg, Germany
(e-mail: {thomas.bangemann, christian.diedrich, matthias.riedl, daniel.wuwer}@ifak.eu).

**Mechanical and Manufacturing Engineering, Loughborough University
Loughborough, LE11 3TU, UK (e-mail: {r.harrison, r.p.monfared}@lboro.ac.uk)

Abstract: While introducing new concepts it is essential to consider the integration of components already existing or being based on alternative and well established technologies. For several reasons like investments, skills of persons, or others it is the usual case not to replace complete installations. These are extended using the same or improved technologies. Within the SOCRADES project, the integration of non web service enabled automation components is done using gateways or mediators. The integration of this kind of automation equipment also imposes requirements regarding extended configuration capabilities of the SOCRADES engineering concept. The paper presents approaches to integrate existing automation devices into upcoming Web Service based applications. To cope with a broad range of automation equipment and applications, the integration concept targets one of the world-leading industrial communication systems – PROFIBUS.

Keywords: SOA, Web Services in Automation, OPC UA, legacy device integration.

1 INTRODUCTION

The project SOCRADES - Service-Oriented Cross-layer Infrastructure for Distributed smart Embedded devices – investigates the usage of Web Services in the area of automation. The project is funded by the European Community under the grant number EU FP6 IST-5-034116 [1]. In particular the Service Oriented Approach is being adapted to the level as low as the automation devices or facility automation. The outcome of the project is to establish an efficient collaboration between the device-level SOA and the services and applications that constitute the enterprise back-end support. Mainly the adaption of the SOA approach at the device level is demanding, because of its fine-granularity. Furthermore this level is characterised by a high dynamic and is therefore more focused on technical issues rather than on business systems aspects, from where the SOA approach originates. Therefore SOCRADES introduces a middleware level in order to integrate automation devices into the back-end applications, e.g. engineering and programming tools or simulation tools using the services that are offered by devices. Unfortunately there are no such devices available currently. Intention of project is also to demonstrate the approach on a real automation process. Therefore simulation of the interactions is not in scope. Also it is therefore highly desirable to be able to integrate existing automation devices into the SOCRADES approach by means of a special middleware as well as dedicated server concepts.

The remainder of this paper is organized as follows: The second section discusses the internal structure of existing automation devices and extracts the essentials for the integration of such devices in the SOCRADES approach. The third section explains the application architecture from the SOCRADES perspective. The forth chapter presents the produced concept and the fifth section discusses the implemented solution at the demonstration plant. The sixth section gives a perspective on the next steps for enhanced integration. Finally we conclude the paper with a summary of the presented work.

2 STRUCTURE OF EXISTING AUTOMATION DEVICES

2.1 Configuration approach

Digital information computation in field devices and digital fieldbus communication has led to a change in the handling of automation systems in manufacturing and process control. The field devices now contain much more information than the simple 4-20mA signal. In addition, they carry out some functions which were originally programmed within the PLC or DCS. These field devices are also known as smart or automation devices. The following description of a transmitter illustrates the evolution from an analogue 4-20mA device to a smart fieldbus device. Other types of automation devices have also followed a similar development path. Even digital, discrete and I/O field devices are becoming much simpler, for example the 24V technology is being progressively replaced by fieldbus connected devices. In principle, in smart fieldbus transmitters the digital/analogue unit at the end of the signal chain in the transmitter is replaced by the fieldbus controller.
Currently, the measurement value remains digital. Two transformations - digital to analogue in the field device and analogue to digital in the PLC input card - as well as the analogue transmission is skipped. This again increases the accuracy of the signal chain. In addition, fieldbus devices, as illustrated in Figure 1, need specific communication commissioning using additional fieldbus configuration tools.

![Diagram of smart fieldbus transmitters](image)

**Figure 1:** Structure of smart fieldbus transmitters

The automation device is typically integrated as a component in an industrial automation system. The automation system performs the automation related part of the complete application. The components of an industrial automation system may be arranged in multiple hierarchical levels connected by communication systems. The field devices are components in the process level connected via inputs and outputs to the process or the physical or logical sub-networks [8].

The automation devices are an integrated part of the entire life cycle of control systems starting with planning/design, including purchase, system integration/commissioning, operation and ending with maintenance. During each phase of the life cycle, specific information represented in different formats is used from the tools (functional design, electrical design, commissioning, diagnostic, ...). The planning and design defines the requirements coming from the process design, which defines the type and the properties of the automation device. Commissioning carries out the parameterisation and configuration of smart devices in connection with the programming of the programmable controllers. During operation there are predominantly interactions between the controllers and the automation devices. The smart devices interact with special tools during maintenance. The information view of smart devices, i.e. the used subset of the entire information range, the format of the information (e.g. text in manuals, files, database entries, html pages) and the source of the required information (e.g. paper, machine readable in the system, online from the device) differs between the life-cycle phases. Therefore, integration methods need to deal with multiple technologies.

### 2.2 Information for device instrumentation

Today, the large number of different device types and suppliers within a control system project makes the smart device parameterisation and configuration task difficult and time-consuming. Different tools must be deployed and data in various formats needs to be exchanged between these tools to provide the required integration. The data exchange is not standardised, therefore data conversions are often necessary, requiring detailed specialist knowledge. Therefore, the consistency of data, documentation and configurations can only be guaranteed by an intensive system test.

### 3 AUTOMATION ARCHITECTURE IN SOCRADES

The use of Web Service is a common technology in the business and office domain. Descriptions of this technology are often based on the specific software components within these domains. The following paragraphs show the Web Service approach in the scope of automation systems.

The classical system structure is the central control using the IEC 61131-3 languages in PLCs. Field devices are hard wired to the PLC or connected using a fieldbus. To address the I/O data, the application program uses the directly represented I/O variables, which are typically defined in IEC 61131-3. This represents the control interface of profile based devices.

A decentralised control structure is used in an increasing number of systems where intelligent field devices are connected to a fieldbus. Specific function blocks in the “central” PLC application program, the so called proxy or device function block, are applied. The proxies represent the functionality of the devices and carries out the implicit communication to the device over the fieldbus. In this case the devices are addressed by the input and output variables of the proxy function blocks. This represents both the control and parameterisation interfaces of profile based devices. The parameters are accessed using the communication services such as read or write.

The centralised and decentralised approaches conceal the interactions between the components by centralised control programming using preconfigured communication networks. All function calls are a within single controller and can be instantiated during program compilation time. Additionally the controller has the time trigger i.e. the complete scan over all connected devices.

The Web Service approach focuses on fully asynchronous function calls among the automation system components. The functions are distributed in the system components and the application is a chain of function invocations. A component provides a Web Service for each public function. Therefore a Web Service is a function of an embedded device which can be invoked from unknown remote components (hosts or other field devices). The main difference from the more simple and shorter remote procedure calls (RPC - common in IT technologies over decades) is that the syntax of the Web Services are described and available online in the standardised Web Service Description Language (WSDL) [3]. RPCs specifications are designed specifically for the used protocol.

Web Services are the application layers above the http protocol. It is also a common practice to use SOAP (Simple Object Access Protocol) [4] between the application related Web Services and the http protocol. SOAP provides a large
variety of Web Service capabilities e.g. for routing, service
description, resource access, security resource management.
The main structure of a SOAP message is shown in Figure 2

Figure 2: SOAP-Message overview

SOCRADES benefits from the SIRENA project [7] results,
and more precisely the DPWS (Devices Profile for Web Ser-
vices) communication stack. DPWS is built on top of the
SOAP 1.2 standard, and relies on additional Web Services
specifications, such as WS-Addressing and WS-Policy, to
further constrain the SOAP messaging model. The Profile
defines the following built-in services:

Discovery services (WS-Discovery): those services are used
by a device connected to a network to advertise itself and to
discover other devices and services.

Metadata exchange services (WS-MetadataExchange): those
services provide dynamic access to a device available ser-
vices and to their metadata, such as WSDL or XML Schema
definitions.

Event publish/subscribe services (WS-Eventing): those ser-
vices are extensions of functional Web services, and allow
other devices to subscribe to asynchronous messages (events)
produced by a given service.

DPWS defines an architecture that distinguishes two types
of services: devices (hosting services) and hosted services.
Devices play an important part in the discovery and metada-
xchange protocols. Hosted services provide the functional-
ality of the device and depend on their hosting de-
vice for discovery [5].

Other research projects as SIRENA, SOA4D [11] or
OCEAN [12] did not investigate to integrate existing legacy
automation devices into the SOA approach. Especially this
topic is explicit examined in SOCRADES. The results of
this investigation will be discussed in the next sections.

4 INTEGRATION OF LEGACY AUTOMATION
DEVICES IN SOA APPROACH

4.1 Definition of the Manufacturing Demonstrator

The prototype system comprises a Festo Didactic trai-
ning/demonstration rig shown in Figure 3 below, developed
at Loughborough University.

Figure 3: Test Rigs of the demonstrator

The Rig has been split down into components, each of which
will require a network interface for a fully distributed sys-
tem. The components are then sub-divided into elements
which are items such as individual sensors and actuators that
reside on a single component. Each element within a compo-
nent has been described in terms of its functionality and its
input and output requirements. Initially this rig is controlled
by means of a legacy automation device. The components on
this rig are a distribution hopper and a transfer arm. Both
components are controlled by means of binary signals.

4.2 Selection of a legacy automation device

The legacy automation device has to handle 8 digital inputs
and also 8 digital outputs. There is a wide diversity of such
device types on the market. For this example a PROFIBUS
communication system is used. The benefit of this communi-
cation system is the possibility to use both cyclic and acyclic
data communication between the controller and the automa-
tion device. Cyclic communication is normally used for
process data exchange and is the type of data exchange re-
quired for this application. Acyclic data transfer is used for
sporadic data transfer, triggered by the controller or configu-
ration tool. From the point of view of the SOA approach, the
web service invocations are also sporadic data transfers.
Therefore it is useful, if the automation device offers both
communication channels.

In this demonstration a modular electrical terminal CPX
from Festo is used. The I/O module is plugged into a
PROFIBUS controller, see Figure 4.

4.3 Selection of a communication controller

The communication controller has to bridge the gap between
the IT based Web Services of SOA and the communication
system in automation area, especially the PROFIBUS com-
munication system. For the prototype system an Ethernet
gateway ‘isNet Cube’ from ifak system GmbH is used in
combination with a PROFIBUS ‘isNet Pro’ module, see Fig-
ure 5. It offers an easy to configure gateway between Ether-
net based communication and PROFIBUS, supporting both
cyclic and acyclic data transfer on the PROFIBUS communication system.

5 INTEGRATION CONCEPT

5.1 Theoretical considerations

The selected communication controller enables very flexible composition of the whole automation system. This component can be plugged in directly on the rig next to the automation device. The integration with the communication controller is carried out via TCP/IP protocol and therefore the ‘translator’ between SOA and the automation area can be placed on central IT resources in a plant.

The translator is also well defined in SOCRADES [6]. As illustrated by Figure 6, there are two classes of interfaces provided to integrate devices into the Enterprise Production Applications level.

- Gateway: controls a set of lower-level non-service-enabled devices, each of which is exposed by the gateway as a service-enabled device
- Mediator: aggregates various services in DPWS. For instance, a Mediator can be seen as a Gateway except that it could hide (or surrogates) many devices.

Mediators support functionality beyond gateways since they introduce semantics in the composition. Mediators aggregate, manage and eventually represent services based on some semantics.

Figure 6 shows also an abstract Mediator box which provides Web Services with pre-processed device parameters and functions. This provides added value to the host applications.

5.2 Practical considerations

The integration concept deployed for the SOCRADES prototype system is shown in Figure 7. It is customised for the integration of the Festo CPX I/O device with a PROFIBUS communication interface.

The recent implementation offers high-level interfaces for getting the inputs and setting the outputs of the automation device. The implemented application controller (not visible in Figure 7) uses this interface in order to interact with the automation process. The Mediator runs on a normal PC and is implemented in Java. Therefore it could also be invoked on different operating systems. The interaction between the Mediator and the PROFIBUS controller shown in Figure 7 is done via TCP/IP. This lower level interface of the Mediator to the PROFIBUS controller is very easy to expand and only acts as a PROFIBUS consumer.

In the initial implementation, the interaction between the PROFIBUS controller and the Festo CPX has an acyclic characteristic. Thus the CPX terminal must be configured before it can accept commands from the acyclic channel. This is a typical function needed in the commissioning phase. The required parameters to modify the device behaviour are well documented and could be transformed into an Electronic Device Description (EDD) according to IEC 61804-3 [9]. Within this standard, PROFIBUS is defined as a particular profile. By means of the EDD, the engineer can interact with the device at the parameter level, without knowledge of the physical communication. The EDD will be interpreted at runtime in an interpreter, providing the appropriate user interface and the communication adapter.

All features of EDD, especially the graphical representation of device parameters, are not used inside the example EDD for the Festo CPX.

As mentioned before, the Mediator is specifically designed in order to interact with acyclic communication services. The Mediator supports also the task of pre-configuration of the Festo CPX terminal to accepting such acyclic commands. This is done implicitly, when the Mediator starts up.
One of the discussed approaches in SOCRADES for interaction on Web Service level is also OPC UA [10]. Primarily designed for commissioning and diagnostic reasons, EDDL can also be integrated into OPC UA servers. In this approach, EDD configures the OPC UA server for the interaction with a specific device type, e.g. existing parameters of the device, its user relevant label or relationships between the parameters (business rules).

In SOCRADES such an OPC UA server is being developed in order to integrate legacy PROFIBUS devices into SOA based applications. Therefore, the OPC UA server plays a role of a generic adapter for the integration of most legacy devices, providing a PROFIBUS communication interface.

Thus the EDD-interpreter is integrated in the OPC UA server directly. By means of the EDD (an instance describing a real device type), the OPC UA server can provide data items of the device parameters. The user can use any OPC UA client and can interact via the OPC UA server with the legacy field device.

Figure 6: SOCRADES device integration via Gateway or Mediator

Figure 7: Mediator for Integration of PROFIBUS devices

Figure 8: User interface described in EDD

6 FURTHER DEVELOPMENT

Figure 9: OPC UA server for PROFIBUS devices
Figure 8 shows the internal structure of the OPC UA server. The OPC UA server provides information for all devices, defined in the configuration information. In a first step, this configuration will be done by means of a separate configuration file, e.g. with XML syntax. It provides information about all offered devices with its PROFIBUS addresses and its device type. The definition of the device type is the relationship to the EDD file to be used for the existing device.

Due to the use of the OPC UA server for commissioning reasons, only PROFIBUS DP-V1/MS2 datagrams (acyclic read / write) are necessary and will be exchanged.

The next step is the integration of the OPC UA server into the mediator concept of SOCRADES: Here the mediator will be enhanced by an additional interface as shown in Figure 10.

Figure 10: Integration of OPC UA in Mediator

The Mediator transforms the DPWS communication facilities into the lower level communication of the PROFIBUS, using DP-V1 communication. Depending on the involved real devices and the offered service of the Mediator, additional logical pre-processing inside the Mediator has to be modelled.

7 SUMMARY

The paper has presented an approach to the integration of existing legacy devices into Web Service based applications, as realized in the SOCRADES European IP SOCRADES project. SOCRADES utilizes Web Services based on DPWS and OPC. The presented solution offers considerable potential in allowing Web Service based applications to be practical adopted, because the existing devices at the automation level can be integrated very easily.

Web Services need syntax and semantic specifications to be used in a system. Thus a specific component, defined as a Mediator, is defined in SOCRADES. At its Web Service interface level, only syntactical and semantical well defined interfaces are provided. The complex handling of a legacy device is completely hidden inside the Mediator.

In order to reduce the effort in the implementation of such a Mediator, the mediator can also interact with an OPC UA server. The OPC UA server can be configured for several devices and can be adapted to the different device types by means of device descriptions, based on IEC 61804-3. Thus a very wide range of device types and applications can be covered by means of the presented approach.

ACKNOWLEDGMENT

The authors would like to thank the European Commission and the partners of the European IST FP6 project SOCRADES (http://www.socrades.eu/) for their support of this work.

REFERENCES