Technologies for SOA-based Distributed Large Scale Process Monitoring and Control Systems

Francois Jammes[‡], Bernard Bony[‡], Philippe Nappey[‡], Armando W. Colombo[‡,#], Jerker Delsing[*],
Jens Eliasson[*], Rumen Kyusakov[*], Stamatis Karnouskos[§], Petr Stluka[†], and Marcel Tilly[μ]

[‡] Schneider Electric, France. Email: {francois2.jammes, bernard.bony, philippe.nappey}@schneider-electric.com,
[#] University of Applied Sciences Emden & Schneider Electric, Germany. Email: awcolombo@et-inf.fho-emden.de
[*] Luleå University of Technology, Sweden. Email: {jerker.delsing, jens.eliasson}@ltu.se
[§] SAP Research, Germany. Email: stamatis.karnouskos@sap.com
[†] Honeywell, Czech Republic. Email: petr.stluka@honeywell.com
[μ] Microsoft, Germany. Email: marcel.tilly@microsoft.com

Abstract-In a SOA-based system the applications are organized in a manner such that interoperable services can be used from different domains. In a process industry context, different domains can refer to, for example, process instrumentation and monitoring, execution of process control, data acquisition, etc. Large process industry systems are a complex and potentially very large sets of multi-disciplinary, heterogeneous, networked distributed systems. Current industrial process control systems are typically vendor specific; in addition the different domains are associated with different layers, different standards and different technologies. In the paper the authors report about the investigations and assessments performed to find answers for four major critical questions that arise as key when technologies have to be selected and used in a true Service Oriented Architecture (SOA) based distributed large scale Process Monitoring and Control system: (1) Real-time SOA (what are the limits of bringing SOA into high performance control loops?); (2) Management of large scale industrial distributed control systems (is it feasible to manage up to tens of thousands of service-oriented devices?); (3) Distributed event-based systems are asynchronous (what are the limits compared to traditional periodic scanning systems?) and (4) Service specification (which semantics are the most suitable for specifying process control and monitoring services?).

Keywords— Service-Oriented Architecture (SOA), Web Service (WS) Technologies, Large Scale Distributed Systems, Process Monitoring and Control.

I. INTRODUCTION

Current industrial process control and monitoring applications are facing many challenges as the complexity of the systems increases and the systems evolve from synchronous to asynchronous. When tens of thousands of devices and systems are Service-oriented, asynchronously interconnected and share and exchange data and information, i.e., services, for monitoring, controlling and managing the processes, key challenges such as interoperability, real-time performance constraints, among others, need to be achieved. The SOA-based approach proposed by the European R&D-Project IMC-AESOP (*ArchitecturE for Service-Oriented*

Process - Monitoring and Control, see http://www.imc-aesop.eu and [1]) is addressing some of those challenges. As depicted in Figure 1, the industrial process environment is mapped into a "Cloud of Services", i.e. devices and applications distributed across the different layers of the enterprise are exposing their characteristics and functionalities as "Services". On the other side, these devices and systems are able to access and use those "Services" located in the cloud [11].
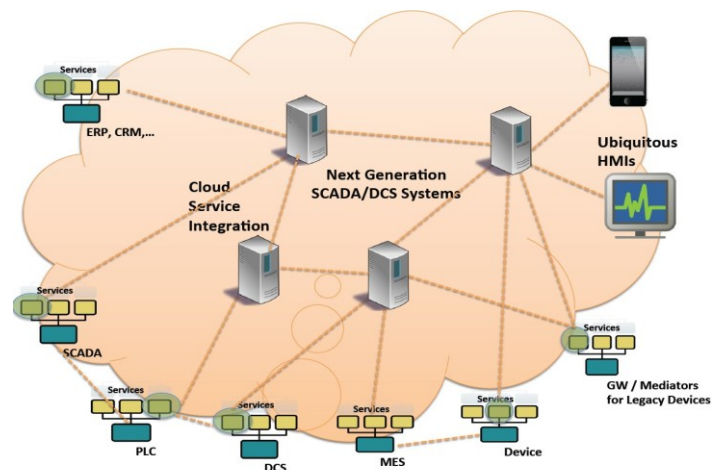

Figure 1. IMC-AESOP Approach. A Distributed Dynamically Collaborative Service-oriented SCADA/DCS System

The outcomes of the first set of technological industry investigations and pilot application shows that the development and implementation of a Service-oriented SCADA/DCS Cloud present the following, four major challenges:

1. Real-time SOA: Determine the real-time limits of bringing SOA inside the high performance control loops of process monitoring and control (e.g. is it possible to provide service-oriented solutions targeting the one millisecond performance range)?

2. Large scale distributed process control and monitoring system: Is it feasible to dynamically design, deploy,

configure, manage and maintain an open plant/enterprise wide system, with thousands of devices and systems operating under process real-time constraints (See standards for Enterprise Architectures, e.g., in [2] and [3])?

3. Process Monitoring and Control Systems operating in an asynchronous mode, e.g., Distributed event based systems: Which are the technological consequences and limits of those asynchronous SCADA/DCS platforms when compared to traditional implemented periodic systems? Is it possible to integrate asynchronous and synchronous systems, e.g. for legacy system integration?

4. Service specification: Which methodology and tools are the most suitable to identify and specify the semantics for interoperable (standard / common / specific) web services based monitoring and control (from business process to devices)?

Along the paper the authors report about the investigations and assessments performed to find answers for those four major critical questions when technologies have to be selected and used in a true Service Oriented Architecture (SOA) based Distributed Large Scale Process Control and Monitoring System.

This paper is organized as follows: after the Introduction, Section II presents a description and assessment of the most suitable technologies, for addressing the 4 challenges described above. Section III presents the results of the assessment synthesizing the technologies that are being used to implement the IMC-AESOP approach. Finally, Section IV concludes the paper and highlights some outlooks.

II. TECHNOLOGY FOCUS AND ASSESSMENT

In regard of the four main challenges addressed in Section I, several technologies have been identified as the major candidates for being used to develop such a Cloud of SCADA-/DCS-Services.

Note: Establishing an exhaustive list was not looked for (it would probably be impossible to achieve), but the major intention of the authors is to offer a compilation/screening of suitable SOA-based technologies, selected following the following main criteria:

1. The technology trends reported in the most recent available publications in conferences and journals;
2. The technologies that are proposed as outcomes of on-going standardization activities;
3. The potential industrial availability at short term either as open-source solutions and/or supplied by the IMC-AESOP technology-provider partners;
4. The originality and innovation associated to the technology:
5. The potential use of a technology by the end-user industry.

A. DPWS / EXI

DPWS [4] [15] [16] is recognized as a very good SOA device level protocol profile. Among all Web Services protocols, it selects the most appropriate ones, such as WS-Discovery and WS-Eventing above SOAP, for implementation in constrained embedded devices. It provides capabilities such as interoperability, plug and play, and integration capability (see http://www.socrades.eu and http://www.sirena-itea.org). However, it does not provide alone real time performance in the millisecond range, as shown in [13]. Associated with EXI [5], this performance target is achievable:

When looking at the real time challenge, the performance that is evaluated and measured is defined as the time to send an event from one device application to another remote device application on a local network. This is done taking into account the time periods required to go through emitter and receiver stacks and to go through the local network, in a one way asynchronous event transmission.

In the example shown in the Figure 2, two remote devices are connected by a physical network (e.g. Ethernet). The first device is detecting a data change on one of its physical inputs, and is sending this information through the network to the second device, which then generates a corresponding physical output. Both devices are using DPWS [4] in order to exchange the information, which provides all the customer values of DPWS (interoperability, plug and play, integration capability). They integrate inside the DPWS stack the EXI encoder or decoder capability in order to add real time performance to the standard DPWS values.
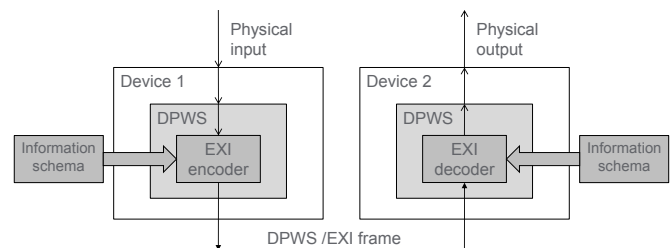

Figure 2.   DPWS / EXI Integration

After this exchange, the first device, when receiving an input change, will translate this physical event into a DPWS / EXI network event, using the combined capabilities of the DPWS stack and of the EXI encoder, which was programmed or configured according to the information schema. The second device, when receiving the network event, will decode the frame and transform it into an output change.

The assessment of the combined DPWS / EXI solution shows impressive results, both in term of frame size compression (between 4 and 18 at run-time, depending on the content) and of performance improvement (between 2.5 and 4 at run-time).

Note: The achievement demonstrates that this SOA asynchronous solution can be at the same performance level

as traditional periodic scan-based solutions used by field buses.

B. EXIP – EXI Project

The implementation of XML/EXI technology (see [5]) provides a very generic framework for describing, implementing, and maintaining complex systems and interactions. However, the usage of XML, even when used with a binary compressed representation, can result in a too high overhead for deeply constrained devices. Furthermore, the application of complex schemas and WSDL descriptions can make versioning difficult since the XML/EXI parsers might require updated grammars for optimal performance.

In some cases, the use of a more simple data representation such as JSON and SenML might be sufficient, especially for very low-cost sensors and actuators. However, the implementation of different data representation techniques between the resource constrained devices and more capable systems requires service gateways that convert these data formats. Using service gateways and mediators introduces complexity in the provisioning and maintenance of the systems. In such scenario it is beneficial to use EXI all the way down to the sensor and actuator devices.

Although the EXI format is designed for high compression and fast processing, its deployment on deeply constrained devices such as wireless sensor nodes is challenging due to RAM and programming memory requirements. In [6] the authors presented the EXIP open source project that is targeted at providing efficient EXI processing for such embedded devices. The EXIP prototype implementation is specially designed to handle typed data and small EXI messages efficiently as this is often required in process monitoring and control applications for sensor data acquisition.

The EXIP project also includes a novel EXI grammar generator that efficiently converts an EXI encoded XML Schema document into EXI grammar definitions. These grammars are then used for schema-enabled processing which provides a better performance than schema-less mode. This grammar generator enables the use of dynamic schema-enabled processing in constrained environments as the EXI encoded XML Schemes are much lighter to transmit and process. The use of EXI representation of the schemes is possible because the XML Schema documents are plain XML documents and as such they have analogous EXI representation. Working with the EXI representation of the XML Schema definitions brings all the performance benefits of the EXI itself - faster processing and more compact representation.

The use of different XML schemes and even different version of these schemes at run time is challenging. For that reason, an important future work investigation is the support for XML Schema evolutions in the SOA implementations.

Another important aspect is the definition of EXI profile for implementation in industrial environment that will guarantee interoperability and optimal performance of the EXI processing. This profile must specify what options should be used in the EXI headers and how the schema information is communicated between the devices and systems.

C. CoAP

In the era of lightweight integration especially of resource-constraint devices with web technologies, a new application protocol is proposed within the Internet Engineering Task Force (IETF) i.e. the Constrained Application Protocol (CoAP) (see [7] and [14]). CoAP provides a method/response interaction model between application end-points, supports built-in resource discovery, and includes key web concepts such as URIs and content-types. CoAP easily translates to HTTP for integration with the web, while meeting specialized requirements such as multicast support, very low overhead and simplicity for constrained environments.
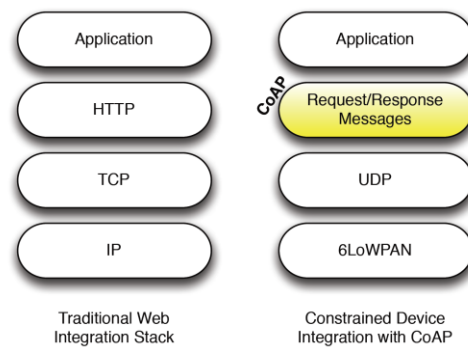

Figure 3. CoAP lightweight integration vs. the heavy HTTP integration

As depicted in Figure 3, CoAP relies on UDP instead of TCP that is used by default for HTTP integration. UDP provides advantages for low overhead and multicast support. CoAP is REST centric (supports GET, POST, PUT, DELETE), and although it can be used to compress HTTP interfaces it offers additional functionalities such as built-in discovery, eventing, multicast support and asynchronous message exchanges. From the security point of view several approaches are supported ranging from no-security up to certificate-based one. IANA has assigned the port number 5683 and the service name "coap".

Within the IMC-AESOP project, CoAP is mainly considered for being used to get access to extremely resource constraint devices, e.g., a temperature sensor, a wireless sensor node, etc. Moreover, the devices may also be mobile and rely on a battery for their operation. These distributed devices would probably be used mostly for monitoring and management, while their integration may enhance the quality of information reaching SCADA/DCS systems.

D. OPC-UA

One of the challenges in process industries is the interoperability between the systems and devices coming from numerous vendors. This has been addressed by using open standards, enabling devices from different vendors to

understand each other. One of the widely accepted standards is OPC (OLE for process control). However, after the years of its use, some limitations of this standard have been evident. This was the reason why OPC Foundation started to work on the new standard – OPC Unified Architecture (OPC UA) [8].

OPC UA main improvements over the classic OPC include the following:

- Unified access to existing OPC data models (OPC DA, OPC HDA, OPC A/E, etc.)
- Multiplatform implementations
- Communication and security (OPC has been based on COM/DCOM)
- Data modelling

While the communication, security and interoperability features make OPC UA great candidate to be used in SOA based applications, it is its data modelling capabilities that enable to build a service oriented process control systems.

OPC UA provides means to access not only the data from the process systems, but also semantic information that is related to the data, like models of the devices that are providing this data. Such models are built by defining Nodes (described by attributes) and Relations between the Nodes.

An information model contains definition of types, from simple to complex, and also instances of such types. The information models are organized and exposed by address spaces. In an existing implementation, multiple information models can be defined, for each level on the process there can be a different model of the process entities, however these models can share some information and usually are synchronized.

With growing penetration of OPC UA to the processes and its features that have been designed with service oriented architecture in mind, it is clear that OPC UA will become a solid part of service oriented distributed control systems.

E. Distributed Service Bus

Web Service based technologies investigated so far (DPWS, OPC UA, etc.) rely on point to point communication models, which do not favour the system scalability.

The "Service Bus" approach aims at decoupling service consumers from service producers in the industrial process control system.

Large scale distributed systems can benefit from a service bus architecture as the bus would act as a broker between the numerous service consumers/providers, avoiding a potentially huge number of point to point connections.

Legacy systems can also benefit from service bus architecture as the bus would act as a gateway between legacy systems and AESOP SOA systems.

This service bus would also be the natural place for adding a semantic layer on top of legacy services. Thus, the bus would provide an abstraction of technical devices and services into business oriented/domain specific services descriptions.

Figure 4 below gives a functional view of such a service bus and illustrates how it could host some of the services identified in the IMC-AESOP architecture study, for instance:

- Gateway functionality through a variety of connectors;
- Registry as a central repository for IMC-AESOP services;
- Code/configuration/model repository;
- Event broker for true loose-coupling between event producers and consumers;
- Security services;
- DNS service;
- Historian/logger;
- Time service for time synchronization between IMC-AESOP services;
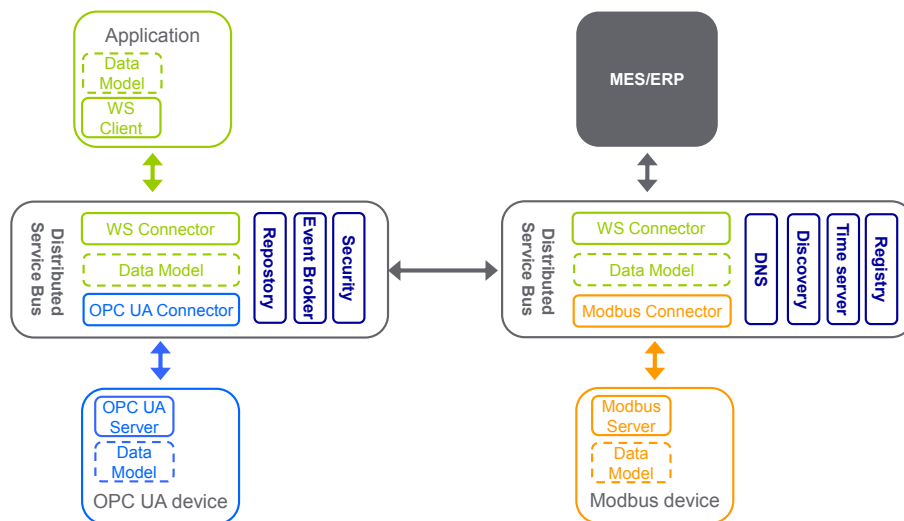- Native interface (Web Services) to higher level information systems (MES/ERP…).

Figure 4.   Distributed Service Bus

F. CEP (Complex Event Processing)

Throughout the last years, Complex Event Processing (CEP) [9] has gained considerable importance as a means to extract information from distributed event-based (or message-based) systems. It became popular in the domain of business process management but is now being applied in the industrial monitoring and control domains. It is a technology to derive higher-level information out of low-level events.

Basically, it relies on a set of tools and techniques for analysing and handling events with very low latency. The feature set for CEP spans from event extraction, sampling, filtering correlation and aggregation to event enrichment, content based routing, event compositions (and not only limited to these).

Normally, complex events are created by abstracting from low-level events. The processing of events is expressed within a specific language in terms of rules. Unfortunately, the set of features and the way to express the rules differ from platform to platform. CEP engines are able to process events up to 100,000 [events/sec]. This clearly depends on the complexity of the rules. Normally the limitation is set by the connection to the external environment, such as extraction of events from input sources or the limitation by the bandwidth of the network.

So far, there is no unified way to express rules (or queries) over streams of events. Thus, it makes sense to wrap a CEP engine within a service with well-defined endpoints. The endpoints are technology agnostic and define the operations and data to be processed while the CEP service itself is responsible for transforming the data/messages to its internal event format. On the output side consumers can subscribe via WS-Eventing so that notifications can be sent via SOAP messages as well (see Figure 5). This approach enables the integration with specifications like Device Profile for Web Services (DWPS) and OPC Unified Architecture (OPC-UA), which are the most suitable solutions for implementing a SoA since both specifications include eventing mechanisms.

Remark: By enabling event processing mechanisms IMC-AESOP is also considering the convergence of scan-based and event-based mechanisms. This is achieved by supporting either pull- or push- models [10]. The services can either send events (*active*) to the CEP service or there is a mediator which pulls data form services (*passive*) and sends this data. From the CEP service this looks like an active data service provider. On the output side results are pushed to registered consumers.
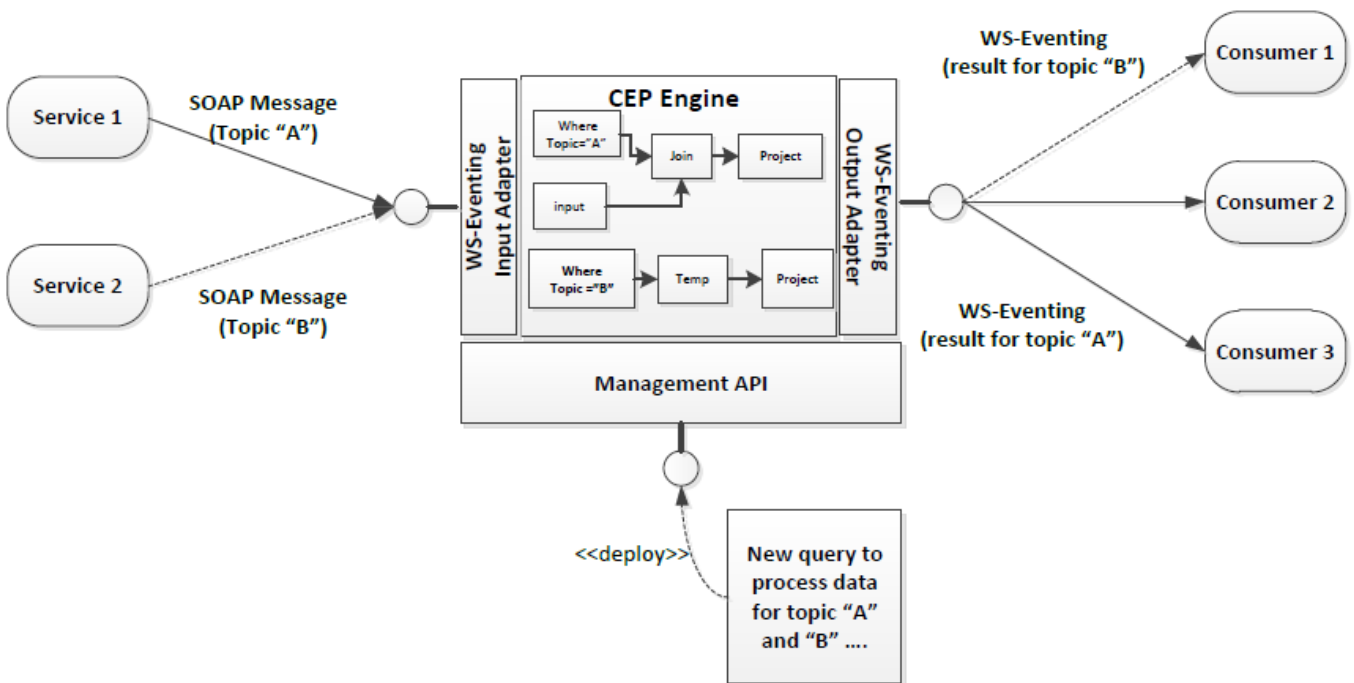

Figure 5. Complex Event processing mechanism in a SOA-Infrastructure

III. Technology selection and synthesis

A synthesis of the most promising technologies, which are currently being used to implement the IMC-AESOP prototypes, is presented in Table I.

TABLE I
Synthesis of technologies being used in the IMC-AESOP Approach

Technologies	Challenges			
	Real-time	Management of large scale	Event driven	Semantics
DPWS		✓	✓	
OPC UA		✓		✓
CoAP		✓	✓	
EXI	✓		✓	
Service Bus		✓		
CEP			✓	

As described above, DPWS, coming from the IT world, is the most applicable set of web services protocols, to be used at the device level. Combined with EXI, it provides real capabilities in the range of the millisecond, following the technology assessment made by the project.

OPC-UA, coming from the industrial world, is also a set of web services protocols, compatible with DPWS, and providing a data model enlarging the semantic capabilities of the solution.

CoAP is the SOA protocol to be used for wireless sensor networks. It can also be combined with EXI.

The Service Bus and the CEP solution are technologies providing the large scale and migration capabilities, combining and processing information coming through DPWS, OPC-UA or legacy protocols, in order to manage large-scale event-based systems.

A suitable combination of the six technologies described above is then able to provide solutions meeting the four critical questions and challenges expressed in Section I.

IV. Conclusions and Future Work

As presented and described along in this manuscript, there are four major critical questions that arise as key when technologies have to be selected and used to implement a Service Oriented Architecture (SOA) based distributed large scale process monitoring and control system: (1) Real-time SOA (what are the limits of bringing SOA into high performance control loops?); (2) Management of large scale industrial distributed control systems (is it feasible to manage up to tens of thousands of service-oriented devices?); (3) Distributed event-based systems are asynchronous (what are the limits compared to traditional periodic scanning systems?) and (4) Service specification (which semantics are the most suitable for specifying process control and monitoring services?).

After compiling and assessing a set of technologies, a subset of them has been selected and presented to the reader as the preferred technologies to be used by the IMC-AESOP consortium. It is important to call the attention to the fact that the selected technologies, presented in Section III, Table I, are either already available from open-source sites or they are still under development by some of the IMC-AESOP technology-provider partners.

The next steps in this on-going research and development work will be the assessment of the prototype implementations, in order to refine the technology evaluation and investigate other challenges in implementing SOA-based cross-domain infrastructures, e.g., cloud of services generated from the virtualization of different systems like manufacturing, smart grid, transportation, etc. [see e.g., [11] and [12].

References

[1] S. Karnouskos, A. W. Colombo, F. Jammes, J. Delsing, and T. Bangemann. "*Towards an architecture for service-oriented process monitoring and control*". In Proc. of the *36th Annual Conference of the IEEE Industrial Electronics Society (IECON-2010), Phoenix, AZ.*, USA, Nov. 2010.

[2] PERA Architecture for Plant-Wide Systems. http://www.pera.net/Pera/Doc_Arch_Articles.html; http://www.pera.net/Levels.html

[3] http://www.isa-95.com.

[4] http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01

[5] http://www.w3.org/XML/EXI/

[6] R. Kyusakov; J. Eliasson and J. Delsing*: "Efficient Structured Data Processing for Web Service Enabled Shop Floor Devices"*. In Proc. of the 20th IEEE International Symposium on Industrial Electronics (ISIE′11), Gdanz, Poland, June 2011.

[7] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, *"Constrained Application Protocol (CoAP)"*, IETF, 12-March-2012, https://datatracker.ietf.org/doc/draft-ietf-core-coap/

[8] W. Mahnke, S.-H. Leitner, and M. Damm. OPC Unified Architecture. Springer, 2009 (http://www.amazon.com/OPC-Unified-Architecture-Wolfgang-Mahnke/dp/3642088422/) :

[9] D. Luckham, The Power of Events*: "An Introduction to Complex Event Processing in Distributed Enterprise Systems"*, Amsterdam: Addison-Wesley Longman, 2002.

[10] M. Tilly and R. Marganiec: *"Matching customer requests to service offerings in real-time"*. In Proc. of the 2011 ACM Symposium on Applied Computing, SAC '11, ACM, 2011, pp.456–461

[11] S. Karnouskos and A. W. Colombo: *"Architecting the next generation of service-based SCADA/DCS system of systems"*. Proc. of the 37th IEEE International Annual Conference on Industrial Electronics (IECON′11), Melbourne, Australia, 7-10 Nov. 2011.

[12] A. W. Colombo: *"Implementing Industrial Cyber-Physical Systems. The Shop Floor as a Service-Cloud?* Keynote at the IEEE ICIT 2012. Athens, Greece, 21st March 2012

[13] R. Hilbrich: *"An Evaluation of the Performance of DPWS on Embedded Devices in a Body Area Network"* Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference 20-23 April 2010

[14] Z. Shelby: *"Embedded web services"* IEEE Wireless Communications journal. December 2010.

[15] G.Candido et al : "Generic Management Services for DPWS-enabled Device » http://www.socrades.eu/Documents/objects/file1259605647.07

[16] S. Karnouskos,et al : "Integration of SOA-ready Networked Embedded Devices in Enterprise Systems via a Cross-LayeredWeb Service Infrastructure" www.socrades.eu/Documents/objects/file1192208072.69