Migration of a Legacy Plant Lubrication System to SOA

Philippe Nappey, Charbel El Kaed, Armando W. Colombo – *Schneider Electric*
{philippe.nappey, charbel.el-kaed, armando.colombo}@ schneider-electric.com
Jens Eliasson, Andrey Kruglyak, Rumen Kyusakov – *Luleå University of Technology, Sweden*
{jens.eliasson, andrey.kruglyak, rumen.kyusakov}@ ltu.se
Christian Hübner, Thomas Bangemann – *ifak e.V., Germany*
{christian.huebner, thomas.bangemann}@ ifak.eu
Oscar Carlsson – *Midroc, Sweden*
oscar.carlsson@midroc.se

Supervision and control systems are being deployed in industrial environments such as mining plants and manufacturing facilities to ensure a continuous and effective production at a minimum cost. Such systems monitor a whole range of devices and collect their data for several purposes like maintenance and control operations.

At the same time, Service Oriented Architecture (SOA) is getting more popular than ever in most application domains, from IT to device level, and particularly for those industries whose continuous efforts to increase the overall plant and equipment effectiveness lead to new requirements on systems openness, integration, availability, maintainability and performance.

This paper presents how a legacy control and monitoring system could migrate to SOA, the architecture principles and the main benefits and limitations based on results validated on a pelletizing plant lubrication system.

I. Introduction

The IMC-AESOP project has been investigating for the past two years how a Service Oriented Architecture (SOA) would benefit to large scale distributed systems in Batch and Process Control applications. The project addresses in particular architectures where large number of service-compliant devices and systems distributed across a whole plant-wide system expose SCADA/DCS monitoring and control functions as Services. More detailed description of the project motivation and high level architecture of SOA enabled process monitoring and control is presented by Karnouskos et al. [11].

One essential investigated aspect has been the co-habitat of currently used synchronous DCS and SCADA with the new asynchronous SOA-based monitoring and control system, going beyond what the current implemented control and monitoring systems are delivering today. We'll detail in this paper the development of an IMC-AESOP demonstrator at the premises of LKAB in Sweden (see Figure 1), implementing an overall control scenario for an existing plant lubrication system and addressing the migration aspects between classical control systems and the new approaches addressed by the project.

Lubrication systems are typical critical systems for almost all process industries. The lubrication control system provides important information that can be used by operators to avoid critical and damaging incidents, by operators, planning staff and management to improve production and plant efficiency and by the maintenance staff and management to analyze and improve the predictive maintenance.

Figure 1: LKAB Plant for Fe-Mineral Processing

The IMC-AESOP plant lubrication use case addresses a number of key points, such as enabling SOA on low level devices, SOA in closed-loop control, integration into an actual plant environment and migration from a scan based PLC to an event based SOA system. In [11], Karnouskos et al. discussed the advantages of using SOA-based solutions for industrial process monitoring and control.

In the following sections, we first outline the existing control and monitoring system then we describe the proposed IMC-AESOP architecture and components. Section IV summarizes the migration aspects into an SOA based solution while section V depicts the main implementation choices. Finally, section VI provides the outcomes of the validation in the real plant facility.

II. Prototype Architecture

The lubrication system, presented in Figure 2, is deployed in the LKAB pelletizing plant [13] on a number of independent black-box systems. Such systems have limited data exchanges between the lubrication systems and the larger distributed control system (DCS). One of these black-box systems will be migrated from the current implementation

978-1-4799-0224-8/13/$31.00 ©2013 IEEE

Similar migration efforts are descibed in the work of Feldhorst et al. in [17] where they use XML/DPWS exclusively as a SOA implementation. In order to extend the service approach to highly constrained embedded devices we propose to use binary encoding for XML and the application protocols which is not investigate in the aforementioned work.

A. Existing system

As shown in Figure 2, the existing lubrication system includes two lubrication circuits controled by a PLC (Programmable Logic Constroler) receiving start/stop commands from a DCS. Each lubration circuit is connected to a pump controled by the DCS through a digital ouput. More than 70 AS-i [14] position switches combined with various digital inputs are scanned periodically by the PLC to get fluid distribution status over each lubrication circuit. Based on this sensors information the PLC controls each pump and directs the fluid to the appropriate circuit. As mentioned above, there is a very limited communication with operational layer, although a touch panel provides a local supervision capability.

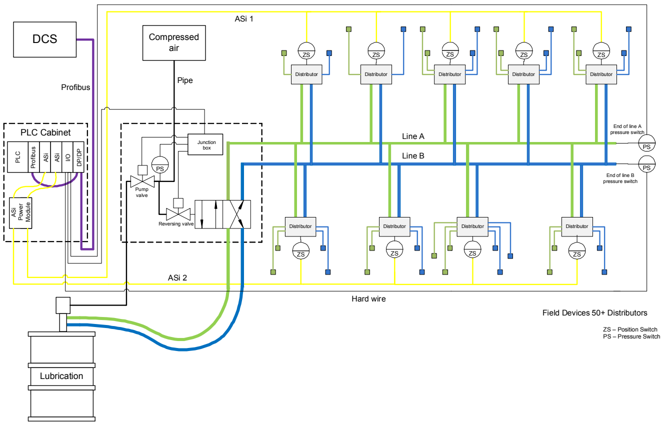


Figure 2: Existing system

B. Proposed prototype

The prototype proposed for IMC-AESOP consists in replacing the existing PLC with an SOA-based system. Thus, the current PLC cabinet is replaced with a SOA-based cabinet and connected to a maintenance station (SCADA), as shown in Figure 3.

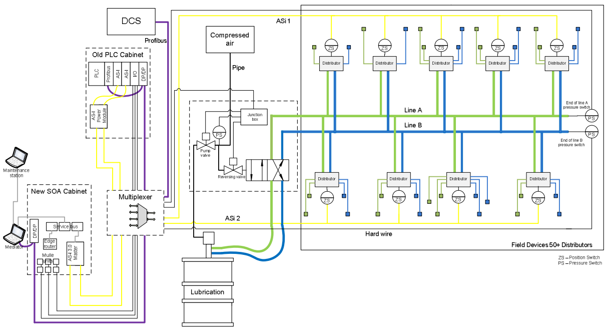


Figure 3: Proposed prototype

III. SOA Components

The proposed SOA architecture is illustrated in Figure 4.

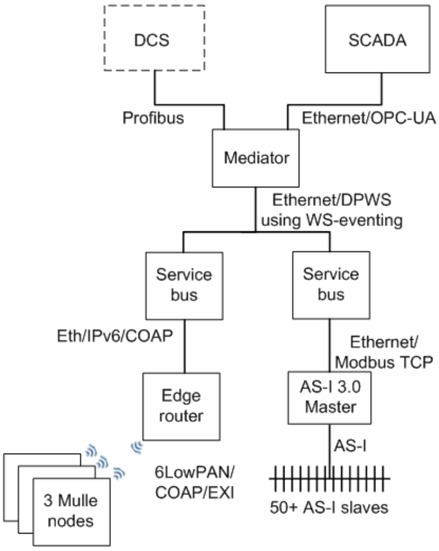
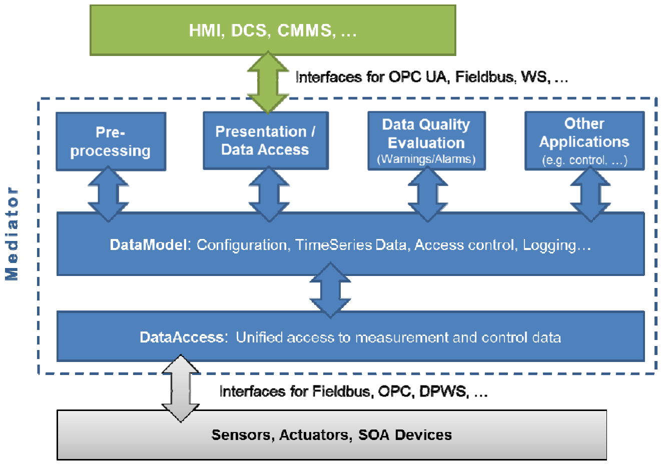


Figure 4: Proposed prototype SOA architecture

A. Mediator

The Mediator provides a runtime system for monitoring and control of process facilities by integrating both legacy as well as SOA-based technologies. It has been built based on an actor-based middleware for fault-tolerant, distributed SCADA systems. The adoption of the actor model [3] for the Mediator implementation results in less complexity and increased reliability compared to conventional (thread-based) approaches to the programming of concurrent processes. As all relevant subsystems are actors that interact with each other only by message passing without sharing common data structures, the actor-based design of the Mediator also greatly simplifies the distribution of parts of the Mediator system. Figure 5 shows the basic structure of the Mediator. The core part of it consists of a data model that describes the logical view of the monitored facilities and also contains all relevant information for acquiring data including communication. The Mediator communicates with the Service Bus through DPWS and also supports basic authentication over SOAP.


HMI, DCS, CMMS, ...
Interfaces for OPC UA, Fieldbus, WS, ...
Mediator
Pre-processing
Presentation / Data Access
Data Quality Evaluation (Warnings/Alarms)
Other Applications (e.g. control, ...)
DataModel: Configuration, TimeSeries Data, Access control, Logging...
DataAccess: Unified access to measurement and control data
Interfaces for Fieldbus, OPC, DPWS, ...
Sensors, Actuators, SOA Devices

For the integration of different communication protocols and information models of various devices and other data sources, an abstract data access layer has been introduced. By providing adapters implemented as actors, any required protocol can be integrated. For the application described in this paper, the PROFIBUS protocol (for connecting to the DCS) as well as the DPWS protocol (for connecting to the Service Bus) has been implemented. In a similar fashion, any processing of the data for pre-processing, control, KPI calculation or presentation to the SCADA HMI layer is easily extendable by providing appropriate adapters.

Within the framework of this SOA system described above, the Mediator data model (including alarms) is presented to the HMI of the maintenance application using the OPC UA protocol.

B. Distributed Service Bus (DSB)

In complement to the Mediator, the Distributed Service Bus provides an additional integration of heterogeneous systems supporting various communication media, protocols, and data models, as shown in Figure 6. Such integration is enabled through loose coupling-based protocol connectors. Each protocol connector reifies devices and services of an existing system into the DSB data model representation. Thus, the Service Bus provides, through a defined abstract layer, a common representation of those devices and services. This abstract layer enables a wide variety of common operations on the underlying systems and devices, including management, diagnostic, maintenance and monitoring.

Moreover, the distribution of the Service Bus provides scalability and evolvability, as each instance can be configured for a specific application domain by implementing dedicated interfaces, quality of service and security requirements. Devices and services handled by an instance of the Service Bus are reified and their information is shared between the other instances through the DPWS protocol.

The distribution feature provided by the DSB is particularly suited to the management of large scale distributed systems, which is central to IMC-AESOP and this demonstrator in particular.

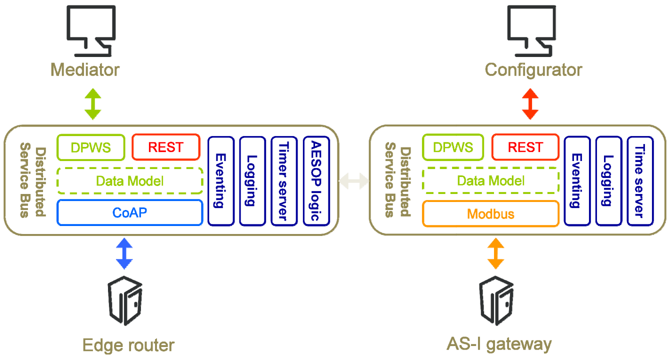


Figure 6: The DSB architecture

C. Industrial Internet of Things

The recent use of internet protocols and web technologies for distributed sensor network installations is gaining wider acceptance [16]. The wireless sensor and actuator network (WSAN), i.e. an industrial approach to Internet of Things (IoT) [10], is built on the 868MHz version of the IEEE 802.15.4 radio standard, which enables low-power communication through thick concrete walls and long-range communication at line of sight operation. The use of Industrial Internet of Things (IIoT) is suitable to use in combination with lightweight embedded systems that are used to measure (and control) the physical parameters of interest.

To make the system scalable and integrate with the IMC-AESOP service cloud, IPv6 was chosen as network protocol. To make the IPv6 network layer comply with the IEEE 802.15.4 Link layer, the 6LoWPAN adaptation layer is used, 6LoWPAN compresses and reduces the data overhead so less energy is required to transfer the information between wireless nodes. IPv6 also enables unique identification of every sensor node using 128bit IPv6-adress. The use of IPv6 also by default includes the network layer security feature of IPsec. Figure 4 shows the edge router which performs translation between IPv4 over Ethernet and IPv6 over 6LoWPAN (IEEE 802.15.4) networks. The edge router also hosts time synchronization services (NTP and PTP) and CoAP services such as data proxy, and also logs the performance of the WSAN. CoAP is a protocol designed for scalability and simplicity [20], whilst being backwards compatible with the much used HTTP protocol.

Mulle [12] devices serve as I/O nodes connecting lubrication pressure switches, air pressure switches, pump valves, reversing valves, and indication lights. Mulle nodes communicates using Efficient XML Interchange (EXI) [4] and CoAP on top of 6LoWPAN. The services hosted by the Mulles supports input, output, filtering, logging, and configuration services. All the data are EXI encoded and transmitted using CoAP over 6LowPAN.

Representing the information measured by the sensors in an efficient yet self-explanatory way is desirable. As the bandwidth in the wireless sensor network is limited, and the energy available in each sensor node is also limited, the efficiency parameter needs extra attention. The concept of service oriented architecture (SOA) is highly interesting in this context as each measured parameter can be represented as a service to the other nodes, but also globally, as the sensors are connected to the internet using IPv6.

D. SCADA

To replace and extend the HMI functionality provided in the legacy system by an integrated touch panel connected to the PLC, a commercially available SCADA solution was used and configured for the use case. The solution used provides a flexible way of presenting data and configuring the system parameters.

Using an OPC UA client, accessing the server provided by the Mediator, the system can be accessed from anywhere on the connected network rather than the current local access only. At the same time the OPC UA server provides a flexible way to access the system with other standardized tools providing a wide array of possibilities.

IV. Implementation Details

The overall prototype was installed in a similar cabinet as the one installed in the plant in order to facilitate the on-site temporary installation of the prototype.

A. I/O nodes

In the demonstration setup, a total of 14 CoAP services (4 actuators, 6 sensors, and 4 outputs used to indicate system status) were implemented. These were located on three nodes, each executing on a Mulle v6.2 from Eistec [6], equipped with a M16C/62P MCU running at 10 MHz and an 868 MHz low-power IEEE 802.15.4 transceiver.

The software on Mulles was implemented in Contiki, with built-in support for CoAP and 6LoWPAN; support for EXI was added to decrease the size of CoAP packets, which allowed us to avoid fragmentation of CoAP packets and improve robustness of communication. The clock of each node was synchronized to the clock of the edge router using NTP. In order to improve the time synchronization performance, the solution proposed by Keunsol et al. [21] is an interesting approach that needs to be further investigated. The complete communication stack is shown in Figure 8.

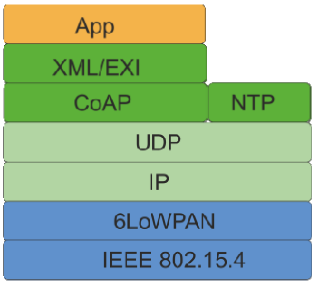


Figure 7: End node communication stack

To enhance the system's security, IPsec is planned to be deployed on the WSAN architecture as well. The use of IPsec on Contiki and 6LoWPAN has been demonstrated by Raza et al. in [22].

B. CoAP and EXI

To implement a SOA-concept in a low-bandwidth 6LoWPAN WSAN used in this application, an efficient compression of the text-based XML service description and data is required. For this purpose EXI (Efficient XML Interchange) [4] was used to represent the XML-based information in binary data format, this reduces the amount of bytes required to represent and transfer the service information. At the application layer, the constrained application protocol (CoAP) is used. CoAP is designed for resource constrained devices like the WSAN nodes used in the demonstration setup.

A key component of the migration of legacy systems to SOA is the use of standard and globally accepted formats for representing the exchanged information. One important result of this demonstration is that it is possible to use EXI for integration of sensor and actuator devices with the SOA automation infrastructure. This enables the implementation of RESTful web services based on CoAP and EXI for industrial

C. Service Bus

The Service Bus has been implemented on two Raspberry Pi devices running Linux operating system and featuring 512 MB of RAM and 700 MHz ARM CPUs. As illustrated in Figure 6 the main software components of the Service Bus are a pivot data format, a set of connectors acting as external interfaces (DPWS, REST, CoAP, and Modbus), an eventing module, a time synchronization (PTP) module, a logging (syslog) module and the AESOP logic which is reproducing the application logic from the existing PLC.

The two instances of Service Bus dynamically discover each other at startup with WS-Discovery and rely on DPWS for message exchanges between them. A basic cyber-security protection is provided by the combination of Role Base Access Control (RBAC) and user authentication mechanisms.

V. Migration Aspects

As proposed in [15] a migration of a large DCS into SOA can be initiated with a smaller step where some key functionality is migrated and the basis of a SOA infrastructure is established in a part of the plant.

This use case provides an example of migration of a number of functional aspects that have been identified in the existing system and provides a minimum requirement of functionality for the SOA-enabled system. Most significant of these are:

1. *Local control loop*

In the existing system local control is performed within the PLC using internal timers and the pressure switches distributed throughout the system to trigger the start and stop of the lubrication pump and activation of solenoid valves.

In the IMC-AESOP use case this functionality has been distributed primarily to the Service Bus, accessing both the CoAP services provided by the Mulle nodes for sensing and actuating and the AS-i sensors data. The main advantage of the SOA design is to provide added monitoring capabilities on the control loop (timers and sensors data are available as services).

2. *Inter-protocol communication*

In the existing system there are only two communication protocols involved: The communication to the DCS is handled through Profibus and the collection of data from a large number of field devices is handled through AS-i.

In the demonstrator several new protocols are introduced as part of the architecture to allow communication within the SOA system, while the existing communication interfaces remain accessible through commercially available AS-i and Profibus master modules, respectively. The conversion between different protocols is handled by the Service Bus and the Mediator, as previously described.

3. *Alarms and warning*

In the existing system alarms are handled through lists of Fault- and Reset-bits with a corresponding list of alarm texts, both in the PLC.

In the SOA solution, those alarms are implemented as events collected from the alarm sources and brokered by the Service Bus. Any interested party can then subscribe to those alarms from the Service Bus. In the demonstrator, the SCADA, the DCS (both through the Mediator) and the Service Bus web client are subscribers of process level alarms. Polling based alarms remain available, which is particularly interesting in a migration context.

4. *Operator manual override and Operator configuration*

Operator manual override and Operator configuration are the two key functionalities provided by the touch panel HMI in the existing system.

In the SOA alternative, the Service Bus is exposing those two functionalities as services that can be called by any (authenticated and authorized) client application. In the demonstrator two client applications are consuming those services: the SCADA (through the Mediator) and the Service Bus web client.

As mentioned before, the loose coupling provided by this approach can be leveraged in future maintenance operations by allowing replacing transparently and independently either the server or the client part of those services.

VI. Validation Results

A. *Functional assessment*

The functional validation of the overall architecture was performed on-site during a scheduled maintenance break of the plant. The IMC-AESOP prototype was connected to the lubrication system, by disconnecting the operating cabinet and connecting instead the SOA cabinet.

The lubrication system was then run for several hours for validating the functional behavior of the prototype and collecting timing data.

B. *Performance assessment*

In order to measure the overall performance of the prototype, the components of the SOA architecture synchronized their time using the PTP protocol (IEEE 1588). All the components were configured to send their logs to a centralized Syslog server (IETF RFC 5424) for timing analysis.

Table 1 below summarizes the average time it takes for a End of line pressure switch event to propagate from the Mulle device to the Mediator through the Edge Router and the Service Bus.

Event	Node	Time offset (ms)
End of line pressure switch	Mulle (sensor)	0
	Edge Router	11
	Service Bus	13
	Mediator	21

Table 1: time measurements

In this example, the CoAP Edge Router receives the event switch, then the Service Bus acknowledges the event 2 ms later and finally the Mediator 8 ms later. The total transmission time between the sensor (Mulle) to the Mediator is 21 ms which is above the current PLC cycle time but stays compatible with the application requirements.

C. *Wireless assessment*

One parameter of interest that is important for successful deployment of 6LoWPAN devices is the size of the messages that the devices must exchange. Using XML is beneficial for integration of the devices with the data models and message formats used in the upper layers of the automation. By using EXI in strict XML schema mode for the low-bandwidth wireless links, the size of the XML messages is reduced more than 20 times. With that, the size of an EXI encoded digital IO process value with timestamp and quality indicator is 10 bytes as compared to 228 bytes for its plain XML counterpart. Another key performance indicator for wireless applications, especially in noise industrial environments, is the occurrence of retransmissions of packets. A retransmission wastes link bandwidth uses energy and increase latency. During the tests, retransmissions were at a low level, with a stable wireless network as a result.

D. *Data modeling*

Enabling interoperability of the service specifications and data models is a key technological challenge that SOA systems are aimed to resolve. The full interoperability requires that the syntax and semantic service descriptions are well defined, unambiguous and enable dynamic discovery and composition. Thus far, most if not all SOA installations are enabling pure syntax interoperability with little or no support for standard based semantic descriptions. The use of structured data formats only partially resolves the problem by supplementing the exchanged data with meta-information in the form of tags and attributes in the case of XML/EXI for example. The tag names are ambiguous and usually insufficient to describe the service functionality in full.

Applying application level data model standards is often used as a solution to that problem as the syntax to semantics mapping is predefined. Example of such standard is Smart Energy Profile 2 that clearly states the physical meaning of the tag names and structures defined for the service messages in the domain of energy management. One problem when complying with such standards is that they are almost always domain specific which requires mapping of the semantic descriptions from one standard to all others in use.

Another approach is to define generic semantic data model that is applicable to wide range of use cases. This is the approach selected for the work presented in this paper. The initial investigation highlighted the Sensor Model Language (SensorML) [18] developed by Open Geospatial Consortium (OGC) as a promising specification for generic semantic description of sensory data. However, the complexity and size of SensorML specification limit its use to more capable devices. Small scale experiments with a number of sample SensorML messages showed that even EXI representation will not be sufficiently small to fit a battery powered wireless sensor nodes that have low-power, low-bandwidth radios.

Another possible specification for sensor data is the Sensor Markup Language (SenML) IETF draft [19]. It has a very simple design that is consistent with RESTful architecture and is targeted at resource-constrained devices. The evaluation of SenML specification showed that it meets the requirements for hardware utilization but there are areas that are too much simplified and insufficient to describe the data in the details required by the target application. Example of such limitation is the precision of the time stamping of the sensor data – SenML allows for up to seconds resolution that is not enough for most use cases. This led to the use of custom generic data representation that is reusing many of the design choices in SenML.

E. *Overall drawbacks and benefits*

A general drawback of the proposed solution is obviously its lack of maturity, in a sense that it consisted in a set of prototypes provided by different partners, none of them being industrialized yet. This translated into both unreliability issues and integration complexity. Part of the integration difficulties consisted in having a specific configuration and monitoring interface for each partner component.

This heterogeneity and relative complexity of the demonstrator can in turn be perceived as an opportunity to validate the SOA approach, each component of the architecture exposing and consuming services to/from other components, with a fairly high level of loose coupling.

In a productized version of the demonstrator, all middleware components (Mediator, Service Bus, Edge Router and potentially AS-I gateway) would ideally be merged into one product, thus reducing the main complexity of the system. However, a projection to a productized version of the SOA middleware would still lead to a higher level of internal complexity compared with a less versatile PLC based solution.

The main benefit of the proposed solution, compared with the installed solution, is to facilitate the overall system installation and maintenance. Although the installation benefit was not obvious on the demonstrator due to the multiplicity of technologies (and partners) involved, the maintenance and monitoring value was fairly obvious thanks to the advanced monitoring capabilities provided by the added services and displayed through the SCADA (timers, sensors values, alarms…).

VII. Conclusion

The on-site validation of the IMC-AESOP prototype provided very positive feedbacks considering that both functional and performance results were in line with customer expectations, combined with added supervision and control capabilities at the SCADA level.

SOA proved to be valuable both at device and application level by providing a high level of loose coupling between the various components of the system. Eventing completed nicely the SOA architecture by reducing the overall latency of the information flow.

On the wireless side, the tests show that CoAP-based services over 6lowPAN can be used for process monitoring More research is needed though in order to improve both scalability and robustness and minimize latency.

Acknowledgment

The authors would like to thank the European Commission for its support through the European Collaborative Research project IMC-AESOP of the FP7 program, which funded this activity.

The authors would also like to thank the LKAB Company for the open collaborative work on this solution and by giving us access to their Kiruna facilities.

References

[1] OASIS, "Devices Profile for Web Services Version 1.1", 2009, http://www.oasis-open.org/committees/ws-dd
[2] SOA4D Forge - Service-Oriented Architecture for Devices, https://forge.soa4d.org/
[3] Carl Hewitt, Peter Bishop and Richard Steiger: A Universal Modular Actor Formalism for Artificial Intelligence. International Joint Conference on Artificial Intelligence, 1973
[4] Efficient XML Interchange (EXI) Format 1.0 W3C, 2011
[5] Kuladinithi, K.; Bergmann, O.; Pötsch, T.; Becker, M. & Görg, C. Implementation of CoAP and its Application in Transport Logistics Proc. IP+ SN, Chicago, IL, USA, 2011
[6] Hartke, K. Observing Resources in CoAP CoRE Working Group, 2013
[7] Kovatsch, M.; Duquennoy, S. & Dunkels, A. A low-power CoAP for Contiki Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on, 2011, 855-860
[8] Kyusakov, R.; Eliasson, J. & Delsing, J. Efficient structured data processing for web service enabled shop floor devices Industrial Electronics (ISIE), 2011 IEEE International Symposium on, 2011, 1716 -1721
[9] Eliasson et al. The Titanium IoT service composition framework. April 2013 http://sourceforge.net/projects/titaniumiot/
[10] Castellani, A.P.; Bui, N.; Casari, P.; Rossi, M.; Shelby, Z.; Zorzi, M. Architecture and protocols for the Internet of Things: A case study. 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010
[11] Karnouskos, S. ; Colombo, A.W.; Jammes, F.; Delsing, J.; Bangemann, T., Towards an architecture for service-oriented process monitoring and control, in IECON 2010.
[12] The Mulle sensor and actuator platform. http://www.eistec.se
[13] http://www.lkab.com/en/About-us/Overview/Operations-Areas/Kiruna/
[14] http://www.as-interface.net/
[15] Delsing, J. et al, Migration of industrial process control systems into service oriented architecture, IECON 2012.
[16] Shelby, Z.; Embedded web services, Wireless Communications, IEEE, 2010, 17, 52 -57
[17] Feldhorst, S.; Libert, S.; ten Hompel, M. & Krumm, H.; Integration of a Legacy Automation System into a SOA for Devices in Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on, 2009
[18] Sensor Model Language (SensorML) Implementation Specification, Standard, OGC, 2010
[19] Jennings, C.; Shelby, Z. & Arkko, J. Media Types for Sensor Markup Language (SenML), Standard, 2013
[20] Bormann, C.; Castellani, A.P.; Shelby, Z; CoAP: An Application Protocol for Billions of Tiny Internet Nodes, in IEEE Internet Computing, vol.16, no.2, pp.62-67, March-April 2012.
[21] Keunsol Kim; Seung-Woo Lee; Dae-geun Park; Bhum-Cheol Lee, PTP interworking 802.15.4 using 6LoWPAN, in Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on , vol.01, no., pp.873,876, 15-18 Feb. 2009
[22] Raza, S ; Duquennoy, S ; Höglund, J ; Roedig, U and Voigt, T,; Secure Communication for the Internet of Things - A Comparison of Link-Layer Security and IPsec for 6LoWPAN, in Security and Communication Networks, Wiley, 2012