

# A SOA-based architecture for empowering future collaborative cloud-based industrial automation

Stamatis Karnouskos<sup>\*</sup>, Armando Walter Colombo<sup>†</sup>, Thomas Bangemann<sup>‡</sup>, Keijo Manninen<sup>§</sup>, Roberto Camp<sup>¶</sup>, Marcel Tilly<sup>||</sup>, Petr Stluka<sup>\*\*</sup>, François Jammes<sup>††</sup>, Jerker Delsing<sup>‡‡</sup>, Jens Eliasson<sup>‡‡</sup>

<sup>\*</sup>SAP Research, Germany Email: stamatis.karnouskos@sap.com

<sup>†</sup>Schneider Electric and University of Applied Sciences Emden/Leer, Germany. Email: awcolombo@et-inf.fho-emden.de

<sup>‡</sup>Ifak, Germany. Email: thomas.bangemann@ifak.eu

<sup>§</sup>Honeywell, Finland. Email: keijo.manninen@honeywell.com

<sup>¶</sup>Prodtec, Finland. Email: roberto.camp@prodtec.fi

<sup>||</sup>Microsoft, Germany. Email: marcel.tilly@microsoft.com

<sup>\*\*</sup>Honeywell, Czech Republic. Email: petr.stluka@honeywell.com

<sup>††</sup>Schneider Electric, France. Email: francois2.jammes@schneider-electric.com

<sup>‡‡</sup>Luleå University of Technology, Sweden. Email: {jerker.delsing, jens.eliasson}@ltu.se

**Abstract**—The last years we are witnessing of rapid advances in the industrial automation domain, mainly driven by business needs towards agility and supported by new disruptive technologies. Future factories will rely on multi-system interactions and collaborative cross-layer management and automation approaches. Such a factory, configured and managed from architectural and behavioural viewpoints, under the service-oriented architecture (SOA) paradigm is virtualized by services exposed by its key components (both HW and SW). One of the main results of this virtualization is that the factory is transformed into a "cloud of services", where dynamic resource allocation and interactions take place. This paper presents a view on such architecture, its specification, the main motivation and considerations, as well as the preliminary services it may need to support.

## I. INTRODUCTION AND VISION

The future industrial automation systems are expected to be complex system of systems [1] that will empower a new generation of today hardly realizable applications and services. This will be possible due to several disruptive advances [2], as well as the cross-domain fertilization of concepts and the amalgamation of IT-driven approaches in the traditional industrial automation systems. The factory of the future will rely on a large ecosystem of systems where collaboration at large scale [3] will take place. This is only realizable due to the distributed, autonomous, intelligent, pro-active, fault-tolerant, reusable (intelligent) systems, which expose their capabilities, functionalities and structural characteristics as services located in a "service cloud". Multidisciplinary in nature, the factory appears as a new dynamic cyber-physical infrastructure, which links many component systems of a wide variety of scales, from individual groups of sensors and mechatronic components to e.g. whole control, monitoring and supervisory control systems, performing e.g., SCADA, DCS and MES functions. The resulting combined systems are able to address problems which the individual components alone would be unable to do, and to yield management, control and automation functionality that is only present as a result

of the creation of new, "emergent" information sources, and result of cooperation, composition of individual capabilities, aggregation of existing and emergent features [4].

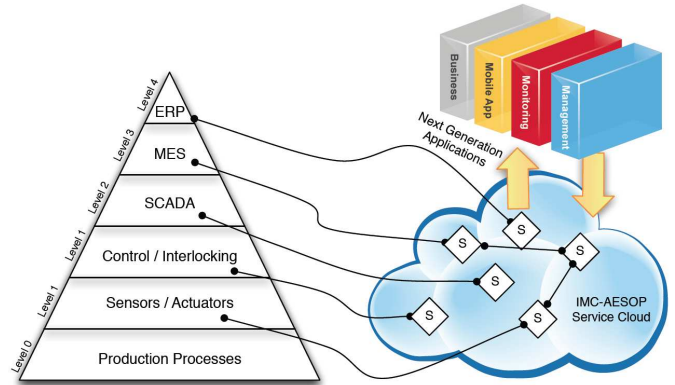


Figure 1. Future industrial system view of cloud-based composition of cyber-physical services

Today, factories are composed and structured by several systems viewed and interacting in a hierarchical fashion following mainly the specifications of standard enterprise architectures [5]. However with the empowerment offered by modern service-oriented architectures, the functionalities of each system or even device can be offered as one or more services of varying complexity, which may be hosted in the cloud and composed by other (potentially cross-layer) services, as depicted in Figure 1. Hence, although the traditional hierarchical view coexists, there is now a flat information-based architecture that depends on a big variety of services exposed by the cyber-physical systems, and their composition. Next generation industrial applications can now rapidly be composed by selecting and combining the new information and capabilities offered (as services in the cloud) to realise their goals. The envisioned transition to the future cloud-based

industrial systems is depicted in Figure 1.

Several efforts so far were 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 standard ([www.isa-95.com](http://www.isa-95.com)). Typically, today's production systems (factory and process) are structured in a hierarchical way in a 5-level hierarchical model. IEC 62264 additionally defines a manufacturing operations management model, implicitly represented by real installations. The standard defines functions mainly associated to level 3 and level 4, objects exchanged and their characteristics and attributes, activities and functions related to the management of a plant, but does neither say anything about the implementations (tools) hosting a specific operation nor the precise assignment to one of the levels 2, 3 or 4. Realizations depend on individual customer needs and the tool manufacturer's strategies. For instance Maintenance Management operation may typically be assigned to a Computerized Maintenance Management System (CMMS), a Manufacturing Execution System – both being typical Level 3 tools – but also to an Enterprise Resource Planning or a Distributed Control System.

Operations can be assigned to specific manufacturing operations management areas i.e. Production Operations Management, Quality Operations Management, Maintenance Operations Management Inventory Operations Management. Having a look into these areas, individual activities can be identified executed within single or distributed source(s). These functions can be implemented using different technologies. Based on these considerations, one can identify distinct directions towards the organizational structure of a production site and the topological or architectural characteristics. From the organizational point of view, the business is structured similar to the levels and operations defined by IEC 62264 - or better to argue in the opposite way: the standard is following what has been developed over the past years.

Topological and architectural characteristics are driven by user or application needs with respect to latest, proofed or acceptable technological capabilities. The major idea is to 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, keeping the organizational aspects established in today's production systems, the migration to future SOA based underlying architecture exploiting the capabilities inherent to SOA is approached [6].

## II. DESIGN CONSIDERATIONS

In order to design the architecture, a set of use-cases and their requirements, as well as concepts and technology trends have been considered. In this section, we focus on the resulting potential directions that may play a key role for the design of the architecture. More specifically, these are:

*Asset monitoring:* The monitoring of assets is of key importance especially in a highly complex heterogeneous infrastructure. In large scale systems [3] it will be practically

impossible to do effective information acquisition with the traditional methods i.e. often pull the devices for their status. The more promising approach is to have an event driven infrastructure coupled with service-oriented architectures. As such any device or system will be able to provide the information it generated (data, alarms etc.) as an event to the interested entities. Due the close relationship between asset monitoring, control and process monitoring the components required to create a large scale system event-driven architecture are mostly the same; the main difference resides in that assets extend to anything that can create value for the company, and while this includes the machines that are monitored and controlled, it also extends to personnel, material, energy, and to other aspects of the machines that are used in processes.

*Backward / forward compatibility:* The future industrial infrastructure is expected to be constantly evolving. As such it is important to be (i) backwards compatible in order to avoid breaking existing functionality and (ii) forward compatible i.e. feature interfaces and interactions as flexible as possible with possible considerations on future functionality and models to come.

*Creation of combinable services and tools:* The trend in software applications is rapid development of them by combining existing functionality in a mash-up way. It is expected that this trend will also empower next generation industrial applications. Since often the development of such functionality is very much task-oriented, new tools are needed to be developed that ideally can be easily combined in a larger system. Typical examples include the XML Pipeline i.e. connecting XML processes such as XML transformations and XML validations together, Complex Event Processing (CEP) driven interactions, service composition, Yahoo! Pipes etc.

*Cross network dynamic discovery:* For plant operation, maintenance, and engineering, zero-configuration networking can provide the tools for managing a device, or service, throughout its operational lifetime. For instance a new device or system may automatically announce its presence and allow cross-layer optimizations during its operation. The goal is towards real-time awareness of all cyber-physical parts in the network and their capabilities.

*Cross-layer integration & real-time interaction:* The aim here is the optimization at architectural and functional levels of the logical and physical network architectures. To achieve this, several actions must be taken e.g. (i) identify activities and information flows, and their performance requirements (hard real time, soft real time, right time, etc.), (ii) investigate technologies that can be used to meet the identified performance requirements, (iii) determine standard ways for representing non-functional requirements, such as Quality of Service (QoS), and propose solutions where standards do not exist, (iv) determine optimal network infrastructure patterns etc.

*Infrastructure evolution management:* Although industrial infrastructures have up to now been designed for the long run e.g. with 15-20 years lifetime in some cases, in the future they are expected to be more often updated for increased reliability,

take advantage of the latest technologies and provision of new functionality. Being technology agnostic of the future advancements, the main challenge is to be able to design today an infrastructure that will be easy to manage and evolve in conjunction with technology. Better said, how can one design "today" the perfect legacy system of tomorrow? How can today's functionalities be reused and integrated to tomorrow's infrastructure with minimal effort? Typical example scenario is the automatic software update service on all devices in the network, for security and safety reasons. Another example of the infrastructure evolution is the migration as envisioned in the IMC-AESOP project [6]. It is expected that several migration paths will exist, and each of those paths will additionally have its own number and type of migration steps.

*Interoperability and open exchange formats:* As next generation systems will be highly collaborative and will have to share information, interoperability via open communication and standardized data exchange is needed. System engineering of interoperable systems has profound impact on their evolution, migration and future integration with other systems [7]. There are two dimensions of interoperability to be considered (i) cross-level i.e. communication between the various levels of the enterprise system, from the plant-floor level up to the enterprise level [5], with for example systems like ERP or MES; and (ii) cross-domain: the case of multi-disciplinary systems where devices and systems of different domains must communicate.

*Management:* The next generation factory systems will be composed of thousands of devices with different hardware and software configurations. There will be a need to automate as much as possible primarily the monitoring part and also the soft-control of such systems. As such it should be possible to dynamically identify devices, systems and services offered by the infrastructures. It should be possible to do software upgrades and mass reprogramming or re-configuration of whole systems. Additionally (remote) visualization of the real infrastructure is a must, as it will give the opportunity of better understanding and maintaining it. Management of a heterogeneous network of devices and systems is crucial for the feasibility of a cloud-based large-scale architecture. The use of devices and systems from different manufacturers adds requirements such as flexibility and extensibility to a management system. Using a common communication architecture will mitigate some of these constraints. Scalability and robustness are also important factors when the number of managed (SOA-enabled) devices increases. A management system must be able to effectively support tens of thousands of devices with different software and hardware platforms from different vendors.

*Mobility support:* In the factory of the future, the operators are not bound to specialized control centres but will be able to control and monitor the processes in the shop floor using mobile HMIs. This enables access to real time measurements and statistics at any time and location. Mobility support also enables monitoring of mobile machinery (automatic loaders, robots, vehicles etc.). Mobility will need to be considered

towards different angles (i) support for mobile devices e.g. being used as HMIs, (ii) support for mobility of devices i.e. where devices are themselves mobile and the implications of this, (iii) support for mobile users and interaction with static and mobile infrastructure and (iv) support for mobility of services e.g. where services actually migrate among various infrastructures and devices following e.g. user's profile wishes.

*Process monitoring and control:* Although the topology and structure of processing plants are usually fixed, a challenge is still given by the large size of a typical plant, which may have thousands of actuating, sensing and controlling devices. This makes the design, deployment, management, and maintenance of a process monitoring and control system significantly more difficult. A SOA-based approach should address the key challenges in order to provide fine-grained flexible process monitoring and control.

*Provision of infrastructure services:* It cannot be expected that all devices (especially resource constrained ones) and systems will always implement the full stack of software that may assist them in interacting with other systems and their services. As such auxiliary infrastructure services are needed that will enable collaboration of systems and exchange of information. Therefore generic services need to be designed and put in place. This implies (i) assumption about generic services hosted at devices and more complex systems, (ii) generic services provided by the infrastructure itself and assurance that devices and systems can interact with them, (iii) dynamic discovery of additional (customized) services and easy interaction with them. As an example the Infrastructure services should enable (i) peer to peer device/system collaboration and (ii) device to business collaboration.

*Real-time information processing:* We have to distinguish between the technical challenges about hard real-time processing which is about predictive and deterministic behaviour on a device and processing of information with low latency from data sources (e.g. sensors) to the consumer, such as dashboard (or operator in front of dashboard), or database etc. Real-time information processing includes high performance set-ups e.g. in-memory databases, effective algorithms and even potential collaborative approaches for pre-filtering or pre-processing of information for a specific (business) objective and complex analysis of relevant (stream) events in-network and on-device.

*Real-world business processes:* With the standardization and easier integration of monitoring and control capabilities in higher layers, the new generation of business processes can rely on timely acquired data exchange with the shop floor. This has as a result the potential to enhance and further integrate real world and its representation in business systems in a real-time manner. It is expected that the business modellers will be able to design processes that interact with the real world possibly in a service oriented way, and based on the information acquired they can take business relevant decisions and execute them. We consider strong integration with enterprise services among other things, as well as the tuning of a large-scale system of systems infrastructure to the business objectives.

*Scalability:* Scalability is a key feature for large-scale sys-

tems [3] and both aspects i.e. vertical scalability (scale up) and horizontal scalability (scale out) will have an effect on the envisioned architecture and its services. The IMC-AESOP architectural approach following the SOA paradigm on all levels, must support very large heterogeneous networks and their capabilities e.g. ranging from gigabit networks to low-bandwidth, energy-constrained networked sensors and actuators connected over unreliable wireless links. This also implies that the overall network must be able support cross-network interaction with devices that are completely different in terms of processing power, bandwidth and energy availability. An one-size-fits-all approach is therefore not applicable; instead, the proposed architecture must incorporate mechanisms that can manage different types of devices, systems and networks. Recourse availability, Quality of Service (QoS), and load balancing are just a few examples of what the system architecture must be able to monitor and manage.

*Service life-cycle management:* The lifecycle of all parts of the architecture will need to be considered from design phase to the end (i.e. de-commissioning or re-purposing). As this is going to be a highly complex system of systems, tackling the lifecycle management especially of composite (potentially cross-domain) services is challenging. To what extent support needs to rely on the core parts of the architecture and what can be realized as optional extensible add-ons needs to be further assessed.

*System simulation:* Industrial process plants can be considered as complex systems, where a change in one sub-process may result to unexpected consequences in other parts of the plant. Nevertheless, autonomicity of the sub-processes and the sub-systems is needed in order to achieve overall evolution. Therefore a holistic system analysis is needed in order to identify possible conflicts and side effects at an early stage. Simulations of process systems is pursued at different levels with varying detail. It is expected that system-wide simulations will assist in designing, building and operating future industrial infrastructure and their interactions.

*Unique asset identification:* Some kind of standardized universal asset identification and addressing mechanism is required for the architecture to be able to support service-oriented targeted communication between and inside the systems. This addressing mechanism should be flexible, scalable and it should not introduce additional overhead in configuration, performance and complexity. Unique asset identification is very closely linked to the monitoring of assets, as it is a critical part of it. It can stem from the location aspect or from the simple awareness of the qualities and properties of the asset that has to be identified.

### III. A SERVICE-BASED ARCHITECTURE

The IMC-AESOP project follows a SOA-based architecture, a general overview of which is depicted at high level in FMC notation ([www.fmc-modeling.org](http://www.fmc-modeling.org)) in Figure 2. On the left side we see the users who interact with the services (depicted in the middle). The data depicted on the far right side can be accessed with the necessary credentials. Although we consider

that the majority of these services will run on the cloud, some of these may be distributed and run in more lightweight versions on devices or other systems. As long as the SOA-based interaction is in place, they are considered as part of the general architecture view.

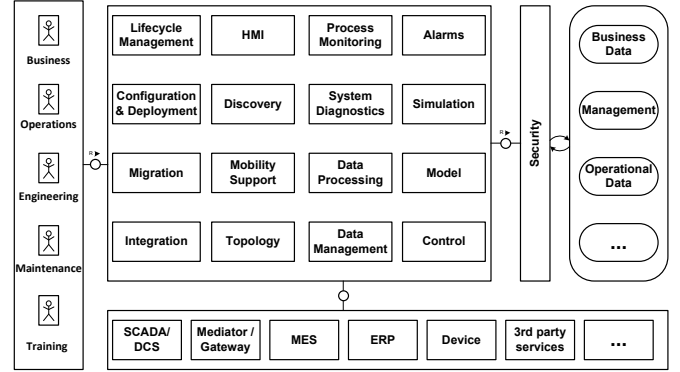


Figure 2. Architecture overview

#### A. User Roles

Several “user roles” are envisioned that will interact with the architecture either directly or indirectly as part of their participation in a process plant. The roles define actions performed by staff and management, and simplifies grouping of task into categories.

The *Business* role handles overall operation, administration, businesses, controls manufacturing and process, etc. This role ensures long-term effectiveness and strategic planning. From IT point of view, this role is operating in the enterprise layer of the process plant, interacting with supporting systems such as Enterprise Resource Planning (ERP), Enterprise Asset Management (EAM), Operational Risk Management (ORM) etc.

The *Operations* role performs the normal daily operation of the plant, hence it handles optimization of the monitor and control processes. It is also responsible for performing efficient, safe, and reliable process operations. The tasks performed as part of this role are located at the operations layer and use supporting systems such as Operations Control System (OCS) for monitoring and control of the process infrastructure and Process Optimization Systems.

The *Engineering* role is here divided into two categories: Process engineering and System engineering. The Process Engineer ensures proper design, review, control, implementation, and documentation of the plant processes. It also designs the layout of the process and performs optimization work with Operations. The System Engineer works with deployment of new devices and machines, manages configuration, the infrastructure and the network.

*Maintenance* is responsible for the system operation with optimum performance, and that the the plant’s systems and equipment are in a safe, reliable, and fully functional state. The maintenance operations are also part of the operations IT



layer of the process plant. The systems that are supporting the tasks performed within the maintenance role are Risk Based Inspections (RBI) systems, Systems Monitoring, Diagnostics and Control etc.

*Training* ensures that all plant personnel have a basic understanding of their responsibilities as well as safe work practices. Training is performed on a regular basis by all other roles in order to improve work skills. The training planning for each employee must be harmonized with the management strategy planning and can be performed on-site but also using Simulation Training Systems.

## B. Service Group Overview

As depicted in Figure 2, it is possible to distinguish several service groups for which there have also been defined some initial services; the latter are listed in more detail in Table I. All of the services are considered essential for next generation cloud-based collaborative automation systems. Table I depicts a first prioritization according to what we consider necessary for future systems. The groups of services have been rated with high priority (+) if they constitute a critical service absolutely mandatory, with medium priority (o) if this is not a critical but nevertheless highly needed service and lastly with low priority (-), which mainly means "nice to have" services that enhance functionalities but are optional.

Table I  
DETAILED ARCHITECTURE SERVICES AND PRIORITIZATION

Service Group	Service	Priority
Alarms	Alarm Configuration	+
	Alarm and Event Processing	+
Configuration & Deployment	Configuration Repository	+
	System Configuration Service	+
	Configuration Service	+
Control	Control Execution Engine	+
Data Management	Sensory data acquisition	+
	Actuator output	+
	Data Consistency	o
	Event Broker	+
	Historian	o
Data Processing	Filtering	+
	Calculation Engine	o
	Complex Event Processing Service	+
Discovery	Discovery Service	+
	Service Registry	+
HMI	Graphics presentation	+
	Business Process Management & Execution Service	o
	Composition Service	+
	Gateway	+
	Service Mediator	+
	Model Mapping Service	+
Lifecycle Management	Code Repository	-
	Lifecycle Management	+
Migration	Infrastructure Migration Solver	-
	Migration Execution Service	-
Mobility Support	Mobile Service Management	o
Model	Model Repository Service	o
	Model Management Service	o
Process Monitoring	Monitoring	+
Security	Security Policy Management	+
	Security Management	+
Simulation	Constraint Evaluation	o
	Simulation Execution	o
	Simulation Scenario Manager	o
	Process Simulation Service	o
System Diagnostic	Asset Monitor	+
	Asset Diagnostics Management	+
Topology	Naming Service	+
	Location Service	+

The *Alarms* service group contains services for alarm processing and configuration. These services support simple and complex events that are composed based on several other

events coming from different parts of the system. Some of the alarms are generated in lower level services and devices, but alarms may also be generated at other layers e.g. in the alarm processing service using process values and limits. The alarm configuration end-processing services also support very flexible hierarchical alarm area definitions and correlations e.g. via heuristics.

The *Configuration and Deployment* service group deals with aspects of overall setup and deployment e.g. ability to handle configuration of (thousands) of devices, configuration consistency validations, deployment of configuration to devices and systems, version management of platform specific implementations of services, instantiation of plant meta-models etc.

The *Control* service group contains the control execution engine service, which is able to execute the process automation configuration or process models. The execution engine services are distributed to several physical nodes and some of those can be redundant. It also supports the typical on-line (and on-fly) changes in configuration while the process is running.

The *Data Management* service group encapsulates the functionality of data retrieval, consistency checking, storage and basic eventing. Data management provides services for acquiring data from sensors, consistency checking and plausibility checks, data logging & searching, event generation, and actuator control.

The *Data Processing* service group provides services from simple filtering up to complex analytics. This is meant in a functional grouping and is intended to be used on all levels, from device up to cloud.

The *Discovery* service group mainly includes services targeting dynamic discovery that allows to find devices/systems/services by type and location and a registry type service, relying on a known registry end-point address, featuring at least register, unregister, search and rating of services operations.

The *HMI* service group contains the graphic presentation service which supports the graphical tools in generic web based user interface framework. It provides the generic menu and help functionalities and also the application area where the actual graphical tools are shown.

The *Integration* service group enables the combination of functionality for added value. Heterogeneous components with different communication protocols and data models require services to facilitate their interoperable interaction. Business process management and execution, composition, functionality wrappers (gateway and mediator), and model mapping services are part of this service group.

*Lifecycle management* is crucial system dealing with the management and evolution of the infrastructure itself. The services provided cover system lifecycle aspects such as maintenance policies, versioning, service management, and also concepts around staging (e. g. test, validation, simulation, production).

The services in the *Migration* service group provide support to migrate a legacy system to a new SOA-based system. This group contains two main services, the Infrastructure

Migration Solver and the Migration Execution Service. The infrastructure migration solver helps identify dependencies and offers migration strategies and instructions. The migration execution service implements migration process according to dependencies and instructions.

The *Mobility Support* service group provides services for managing mobile assets, such as mapping/changing IP addresses, asset locations, tracking, etc. It also provides data synchronization services to enable up to date data access and sharing for mobile services and devices.

The *Model* service group contains services for model management and repository. These services are very generic and those can be used for process automation configurations or process models, and are not limited to these hierarchical models. The model repository takes care of the structure but not the content, hence it is able to support several model types.

*Process Monitoring* services serve as the entry point for the operator through the HMI. It is used to gather information relevant to the physical process e.g. adding semantics to the raw sensor data gathered from Data Processing and Data Management service groups. It also deals with process-related alarms and events.

*Security* is a critical area especially when it comes down to enabling interactions among multiple stakeholders with various goals and access levels. The security management focuses on enforcement or execution of security measures and policy management is about definition and management of security rules or policies. The security services are implicitly used by all architecture services.

The *Simulation* service group is practically related to every other service group in the architecture as it aims at simulation of multiple systems and their processes. It is in charge of evaluating constraints and simulating execution. It also manages simulation scenarios and uses the exposed simulation endpoints provided by other services to emulate the performance and behaviour of a system (or a multitude of systems).

The *System Diagnostic* service group provides features for diagnostics of services and devices. Diagnostics can be used to monitor the health and condition of devices (shop-floor devices, servers, network devices, SCADA systems and PLC, etc.), and status of services. This group is used primarily for maintenance and planning purposes.

The *Topology* service group allows describing and managing the physical and logical structure of the system. It includes Domain Name Service (DNS) functionality, location and context management, network management etc. services.

As we can see these are the main service groups which contain several services each as depicted and prioritized in Table I. While most of these correspond to specific real-world scenarios we consider, expanding the potential scenarios may lead to adjustments on the architecture as such. Within the IMC-AESOP project, several of these are under implementation as proof of concept. There are also several functional requirements which will need to be further evaluated and may depend on domain-specific scenarios. To what extent they

might impact the proposed approach is avenue for further research.

#### IV. THE NEXT GENERATION SCADA/DCS

Service Oriented Architectures are considered a promising way towards realizing the factory of the future [8]. Industrial processes as well as many other critical infrastructures depend on SCADA and DCS systems in order to perform their complex functionalities. The multitude of functionalities that they need to support as well as the exact roadmap is heavily still researched in an environment where disruptive technologies and concepts are developed rapidly [2]. Having in place an architecture as depicted in Figure 2 has profound implications on the design and deployment of future solutions in the industrial automation domain.

Cyber-Physical systems are moving towards an infrastructure that increasingly depends on monitoring of the real world, timely evaluation of data acquired and timely applicability of management (control) [2]. The latter is becoming even more difficult to design and manage when massive numbers of networked embedded devices and systems are interacting. As such new approaches are needed that go beyond the classical monitoring and are able to deal with massive data and complex reasoning depending on the affected processes as well as enterprise-wide constraints. Such "capabilities" would by nature require multi-stakeholder involvement and data access that has to go beyond the classical monolithic one-domain and task-specific development approaches.

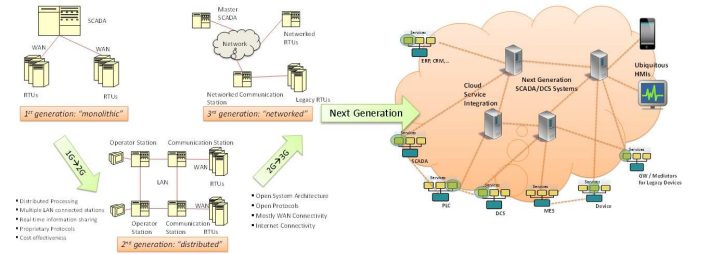


Figure 3. Next Generation SCADA/DCS as a "Services Cloud" [2]

Currently implemented SCADA/DCS systems architectures [9] were designed for more closed and hierarchically controlled industrial environments, however it is expected that there is potential to enhance their functionality and minimize integration costs by integrating themselves into collaborative approaches with enterprise systems and large-scale real-world services [2]. In this sense, there is a need to consider what the next steps could be towards engineering/designing the next generation of SCADA/DCS systems of systems that could successfully tackle the emerging challenges such as degree of centralization, optional independence of each of the participating systems, and independent evolution of them.

The proposed architecture (depicted in Figure 2) could have a significant impact on the way future industrial systems interact and applications are developed. By realizing it, a flat information-based infrastructure (as depicted in Figure 1) that

coexists with status quo is created. This means that the next generation SCADA and DCS systems could heavily depend on a set of common services and strike the right balance between functionality co-located on the shop-floor and delegated into the cloud [2]. The aim is to have an approach that is more fit for the era where the Internet of Things, infrastructure virtualization and real-time high performance solutions are sought. Hence, the next generation SCADA/DCS systems [2] does not necessarily have to possess a physical nature; this implies that it might reside overwhelmingly on the "cyber" or "virtual" world. As such it may comprise of multiple real world devices, on-device and in-network services and service-based collaboration driven interactions mapped into a "Service Cloud" (as depicted in Figure 3).

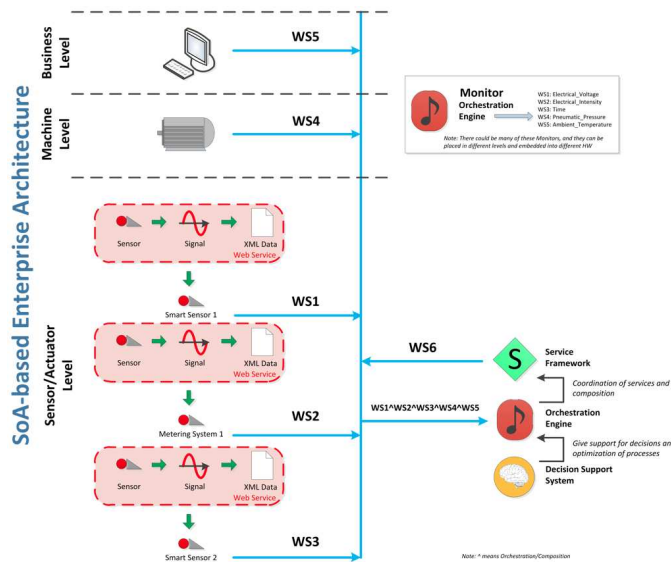


Figure 4. Composition of cyber-physical services

A typical example would be that of asset monitoring with future SCADA systems. In large scale systems it will be impossible to still do the information acquisition with the traditional methods of pulling the devices. The more promising approach is to have an event driven infrastructure coupled with service-oriented architectures. As such any device or system will be able to provide the information it generated (data, alarms etc.) as an event to the interested entities [8].

Following the technological trend depicted in Figure 3, the next generation of SCADA/DCS systems will be having architectural and behavioural characteristics of a SOA-based evolvable infrastructure with the capability of managing emergent behaviours as shown in Figure 4. In this case the existence of a composition or orchestration engine devoted to enable sophisticated monitoring services appears as a new component, with initially unknown geographical position, but it is the result of the necessity to get monitoring indexes in different locations of the enterprise architecture where there are not metering devices or systems able to supply, e.g. energy management services.

## V. CONCLUSION

Future industrial applications will need to be developed at a rapid pace in order to capture the agility required by modern businesses. Typical industrial software development approaches will need to be adjusted to the new paradigm of distributed complex system software development with main guides the collaboration and multi-layer interactions among systems of systems. To do so some generic common functionality will need to be provided, potentially by a service platform hosting common functionalities, following the service-oriented architecture approach. Such a collection of services forming a service-based architecture (shown in Figure 2) is presented, prioritized and their potential impact is analysed. Significant work needs to be invested towards further investigating the interdependencies and needs of all targeted service domains as well as the technologies for realizing them. The proposed service architecture attempts to cover the basic needs for monitoring, management, data handling and integration etc. by taking into consideration the disruptive technologies [2] and concepts that could empower future industrial systems.

## ACKNOWLEDGMENT

The authors would like to thank for their support the European Commission, and the partners of the EU FP7 project IMC-AESOP ([www.imc-aesop.eu](http://www.imc-aesop.eu)) for the fruitful discussions.

## REFERENCES

- [1] M. Jamshidi, Ed., *Systems of Systems Engineering: Principles and Applications*. CRC Press, Nov. 2008.
- [2] S. Karnouskos and A. W. Colombo, "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., Nov. 2011.
- [3] L. Northrop, P. Feiler, R. P. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, D. Schmidt, K. Sullivan, and K. Wallnau, "Ultra-Large-Scale Systems – the software challenge of the future," Software Engineering Institute, Carnegie Mellon, Tech. Rep., Jun. 2006. [Online]. Available: [http://www.sei.cmu.edu/library/assets/ULS\\_Book20062.pdf](http://www.sei.cmu.edu/library/assets/ULS_Book20062.pdf)
- [4] A. W. Colombo and S. Karnouskos, "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), John Wiley and Sons, 2009, vol. 65-81.
- [5] L. D. Xu, "Enterprise systems: State-of-the-art and future trends," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 4, pp. 630–640, Nov. 2011.
- [6] J. Delsing, J. Eliasson, R. Kyusakov, A. W. Colombo, F. Jammes, J. Nesaether, S. Karnouskos, and C. Diedrich, "A migration approach towards a soa-based next generation process control and monitoring," in *37th Annual Conference of the IEEE Industrial Electronics Society (IECON 2011)*, Melbourne, Australia., Nov. 2011.
- [7] R. Drath and M. Barth, "Concept for interoperability between independent engineering tools of heterogeneous disciplines," in *Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on*, Sep. 2011, pp. 1–8.
- [8] A. W. Colombo, S. Karnouskos, and J. M. Mendes, "Factory of the future: A service-oriented system of modular, dynamic reconfigurable and collaborative systems," in *Artificial Intelligence Techniques for Networked Manufacturing Enterprises Management*, L. Benyoucef and B. Grabot, Eds. Springer, 2010, no. ISBN 978-1-84996-118-9.
- [9] D. Barr, "Supervisory control and data acquisition (SCADA) systems," National Communications System (NCS), Technical Information Bulletin 04-1, Oct. 2004. [Online]. Available: [http://www.ncs.gov/library/tech\\_bulletins/2004/tib\\_04-1.pdf](http://www.ncs.gov/library/tech_bulletins/2004/tib_04-1.pdf)