# Analysis of service-oriented architecture approaches suitable for modular process automation

Henry Bloch, Alexander Fay
Institute of Automation Technology
Helmut Schmidt University
Hamburg, Germany
{henry.bloch; alexander.fay}@hsu-hh.de

Mario Hoernicke
ABB Corporate Research
Operations Management
Ladenburg, Germany
mario.hoernicke@de.abb.com

***Abstract*— Evolution in global markets for chemical and pharmaceutical goods requires new approaches in process plant design. Modularization is commonly considered as a promising approach to face the challenges of changing production environment. In order to exploit the advantages of process modularization, new structures of communication and control are required for modular process automation. Service-oriented architecture has been introduced as a promising communication infrastructure combined with state-based control of services. This paper introduces requirements of modular process automation and general requirements of service-oriented architectures. Process functions are represented by encapsulated services in the control level of the automation pyramid. Interaction between a superior control system and intelligent modules is described. Furthermore, approaches of service-oriented architecture are introduced such as a reference model as well as specific implementations of services within modular automation. Subsequently, four different service-oriented architecture approaches are analyzed with respect to their fulfilment of the presented requirements. Finally, this paper discusses the result of the analysis and the general applicability of service-oriented architectures for modular process automation.**

***Keywords—modular process automation, service oriented architecture, state-based control, process module.***

## I. INTRODUCTION

Global markets for chemical and pharmaceutical products are facing challenges concerning shorter product lifecycles and highly volatile markets. A shorter time to market is more and more important [1]. In addition, a forecast of the demanded volumes becomes more difficult because of this volatile environment.

Current production systems are not able to fulfil these requirements without a decrease in product quality or a decrease in output volume. The production systems have to be adapted to this change. A promising approach is the modularization of process plants [2]. The basic idea of modularization is to divide a production system into different modules which each accomplish at least one process function. At the moment this approach is partially implemented in production plants by the usage of package units [3].



The modularization approach, like it has been developed within the F3 Factory project [4], promises further potential advantages by decreasing engineering effort, fast integration of modules and easy numbering up or numbering down in case of changing demand. In summary, modularization has the potential to reduce complexity in engineering, maintenance and during startup phase [5].

A modular plant divides devices, instruments and process control into different segments called modules. Each module fulfills a part of the process and needs to be controlled. Current process control systems (PCS) are designed to control large production systems and are therefore quite inflexible. To control an adaptable and flexible process plant, a new infrastructure of control and communication is required. One approach to make control of modular plants more flexible is described in [5] with a state-based implementation. State-based control follows the recommendations of the standards [6] and [7]. Within each of these standards, a state-based control concept is defined as a control concept with reduced complexity for plant operators. Also a knowledge protection for module suppliers is assured by these concepts. [6] provides an example of a state-machine for batch procedure, whereas [7] determines a variable state model, free to be designed by the user. A possibility to implement new communication infrastructure is by the design of service oriented architecture (SOA) [2].

SOA was developed as an approach for the organization of business processes [8]. It was used in service oriented computing, and also SOA approaches for the domain of automation have been developed [9, 10]. Therefore, it seemed worthwhile to evaluate state of the art SOA-concepts in the context of modular process automation.

Within this paper, a definition of *module* and *modular process automation* is given in chapter II. These definitions are needed because of different understandings in the involved domains within process industries. Requirements of modular process automation are presented in chapter II, as well, followed by SOA approaches in chapter III. To evaluate the approaches, criteria has been developed – from the requirements shown in chapter II and III – which are illustrated in chapter IV. An analysis is conducted in chapter V before a conclusion is made and further research fields are highlighted in chapter VI.

## II. Requirements of Modular Process Auotmation

### A. *Definition of "module" and "modular process automation"*

Regarding modular concepts in the field of process automation, diverse definitions of *module* and *modular process automation* exist. Definitions within process industries reach from a single device to an entire (according to [7]) unit of a plant.

Within mechanical engineering industries, a *module* is defined as a technically and organizationally bounded element which fulfills a defined task [1]. [11] defines *module* as an aggregation of devices and apparatus including their own control system. Modules are suitable for multiple operating points by allowing different overall configurations. If production processes connect several modules, a superior control system is needed.

In the context of this contribution the definitions of the Namur (User Association of Automation Technology in Process Industries, www.namur.net) are used. According to [12] two different module variants exist. The difference between the variants is whether a programmable logic controller (PLC) is part of the module or not. If the module does not include a PLC, the PCS will influence the module directly via a remote I/O. This type of modules can be used for simple process functions. To implement more complex functions or in order to enable the modules for autonomous control of one or more process functions, it is necessary to utilize a module with an integrated PLC.

*Modular process automation systems* are composed of the modules' PLCs and the superior control system. Within the logic programs of the PLCs the intelligence of each module is implemented. Each module is procedurally connected to its predecessor and successor. Furthermore all modules are connected radially to the backbone which supplies and disposes the required media, energy and communication infrastructure. The superior control system, which includes the orchestration functionality of the modules and an interface to the operator, is also connected to the backbone [12]. This infrastructure provides two levels of control: a) basic automation and basic process functions of each module are controlled by the module's PLC; b) recipe management, more complex functions as well as the human machine interface (HMI) are embedded in the superior control system [12].

Referring to [2] the current PCSs do not provide the infrastructure and functionality as needed to achieve the required flexibility, adaptability and evolution of production systems. Therefore, a new communication and control infrastructure is needed.

### B. *Infrastructure and requirements of modular process automation*

The ZVEI (German Electrical and Electronic Manufacturer's Association, www.zvei.org) suggested an infrastructure for modular process automation following the recommendations of [12]. According to this recommendation, a PCS is a required element of the infrastructure, because the configuration of multiple modules can be seen as *plant area* defined in [6], which has a size and complexity that can only be handled by its own control system. To control each module by itself, a state-based control is suggested. Therefore each module must transmit a description of its states to the PCS. The PCS controls the modules via state changes [5]. A possible state-model (Fig. 1) is proposed in [7]. This state-model is an example within the standard; extensions to this state-model are possible.

Beside the infrastructure for control, a new functional structure for modular process automation is required. Therefore the provision of services as encapsulated process functions is suggested by [5]. Each service fulfills a part of the control function within a module. The granularity of each service is chosen freely. Fine granularity results e.g. in an intelligent pump service, whereas rough granularity could be chosen for a service of mixing different educts in a specific ratio, which makes use of the pump service and several other services. In order to use abstract descriptions of the services, an encapsulation of the function and each service requires its own state machine [5].

The change of the communication infrastructure and control structure by modularization of process plants should not affect the functional operability of the PCS. The functionalities and operator interfaces of the plant should be kept as "classical" as possible in order to keep change barriers for stakeholders of process industries low and to gain acceptance within the domain.

Typical functions of a PCS as mentioned in [13] are:

- Provision of a HMI to influence the technical process
- Control and adjustment of the technical process
- Provision of recipe/batch-interface
- Supervision and alarm management
- Archiving of parameters and process data
- Evaluation of data
- Recording of signals

Modular plants require possibilities for testing and verification of the single modules as well as of a group of modules. Test of single modules will be performed by module suppliers within a suitable test environment, whereas the testing of the module network will be done by the module integrator. Subsequently to testing the module network has to be piloted and then monitored during runtime. All new communication and control structures need to enable these specific requirements.

In order to implement a state-based control with encapsulated services, the information which is generated within the modules is required in the PCS in an appropriate way to display it on operator screens and to execute control and to steer the chemical process [14]. Each service should be controlled via state changes triggered by the PCS. A response of all services of each module is required. The response has to

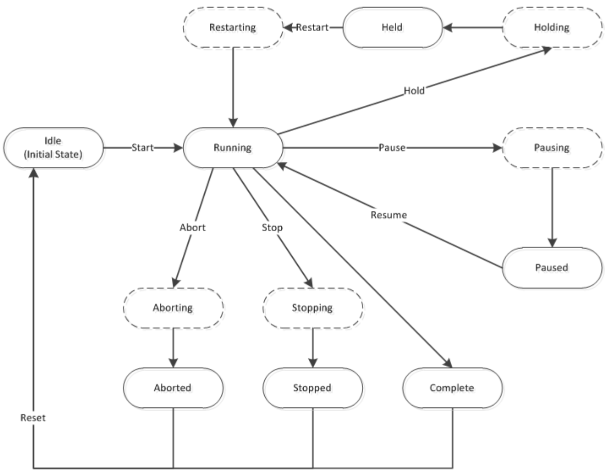


Fig. 1. State-model referring to [7]

contain the active state of each service, the planned remaining time of the active state, and possible following states and by which commands they are accessible. Optional also the active parameters and the next planned state could be responded. The source of the commands is the PCS, which has to orchestrate all services of all modules to an overall process. The communication of the commands and responses can be implemented by an adapted phase logic interface.

These requirements are prerequisites for a state-based control of modular process automation. SOA approaches could fulfil these requirements, which is shown in the analysis within chapter V.. In the next chapter, state of the art SOA approaches and related requirements are introduced.

## III. SERVICE-ORIENTED-ARCHITECTURE APPROACHES

### *A. General SOA Requirements*

SOA was originally an organizational principle based on business processes [8]. It was further developed and used for different domains. The advantage of SOA is to organize large networks, which require interoperability, by a simple paradigm. It is also scalable and evolvable [15]. The use of SOA within an automation environment is not a new approach. Epple introduced in 2008 a model of an architecture which shows how functionalities of each level of the automation pyramid can be implemented as services except of the process level (Fig.2) [16].

In order to use SOA in an industrial environment and especially in modular process automation, a common understanding of SOA is required. In general, SOA describes a set of black box components which provide services via an interface that is implemented using communication services [17]. Within the context of modular process automation, the black box components represent the modules, fulfilling process functions by encapsulating services.

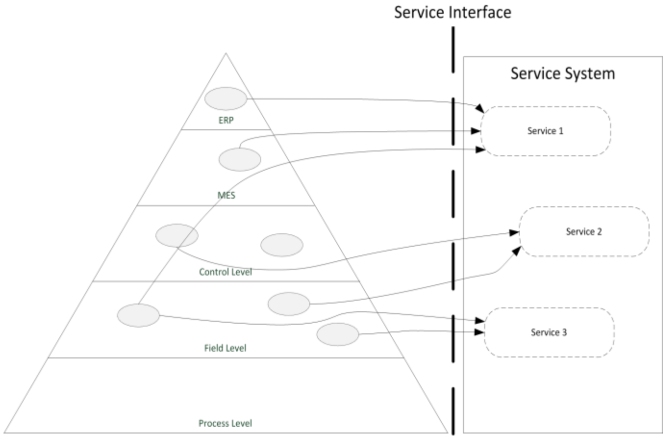


Fig. 2. Concept of service system and automation pyramid referring to [16]

The World Wide Web Consortium gives a definition of a service as a web service which consists of a set of components that have an interface description, are published, can be discovered and executed [18]. This definition fits also on abstract level to modular process automation, because process functions are able to be invoked and discovered by the PCS and operate after invocation.

Below, some SOA approaches are discussed. At first, the general reference model OASIS is introduced, afterwards SOCRADES as an approach from factory automation.

Later on, two further approaches are discussed which use well known modelling languages, such as Petri nets and function blocks, to implement SOA in the domain of automation. The choice of the discussed approaches raises no claim for completeness, but outlines the applicability of SOA approaches in the context of the automation of modular plants.

### *B. OASIS Reference Model*

In 2006, the OASIS reference model was published for SOA. Within the reference model a service is defined as a "…mechanism to enable access to one or more capabilities…" [15]. Services are provided by an entity and are used by other entities. Access is implemented via a service interface, following the pattern of request-response. Each service can have a real world effect. OASIS defines three fundamental concepts (Fig. 3) involved in an interaction with services:

*a) Visibility*

*b) Interaction*

*c) Real world effect*

a) In order to perform an interaction between service provider and service consumer, both need to "see" each other. Therefore, the requirements of awareness, reachability, and willingness have to be fulfilled. Awareness describes the knowledge of both, service provider and consumer, about the existence of the other party. To achieve this, service descriptions are needed in order that the consumer knows if the service is able to fulfill the required task [15]. In the context of

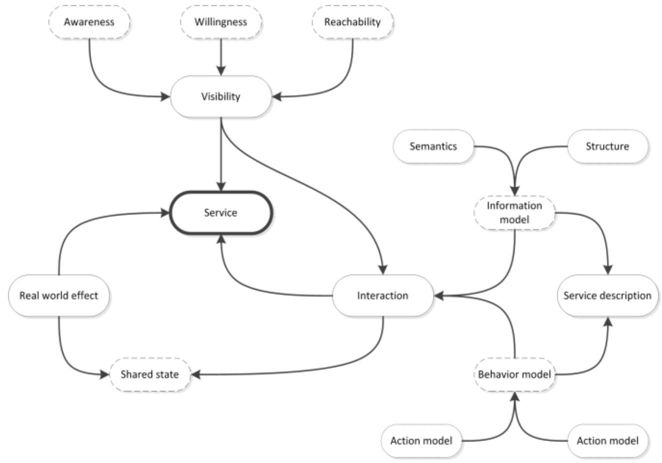


Fig. 3. Concept of Visibility, Interaction and Real world effect referring to [15]

modular process automation, a task is a process function which is operated by a service of a module. The service description is delivered for all services of each module en bloc in the module type package (MTP) [14]. Besides being aware of the services and the involved parties, each party has to be willing to interact. Willingness describes that each involved entity wants to cooperate. In the context of modular process automation, willingness is assumed in general, but it is conceivable that a service cannot be executed because of functional safety requirements or restrictions of other services. Another requirement besides awareness and willingness is that services need to be reachable in order to be visible. Reachability within the OASIS reference model describes the possibility of communication between service provider and consumer [15]. Reachability is assumed in the context of modular process automation by the network and the communication protocols, e.g. OPC-UA.

b) The concept of interaction includes the service description and two main models. On the one hand, the information model characterizes the possible data which can be exchanged among the service provider and a consumer. The model is separated into a structural and a semantic part. The structure defines the type and the form of the data in different structural levels [15]. The semantic part of the information model describes all necessary information for the interpretation of the data. On the other hand, the behavior model contains all knowledge about the actions fulfilled within each service. Also the process' temporal aspects are part of the behavior model. This information is stored in two more detailed models: action model and process model. All specific actions executed in services are defined in the action model [15]. In the context of modular process automation, actions are commands to actuators per service. It does not matter how a command is transferred to each actuator, it is only important that it has been transferred. The process model contains all relationships and properties of each action. This process information is of temporal nature and changes by each action change. The term "process" in this context is meant as the execution of an action and not as the technical process within process industries. Within the reference model, the process model also provides space to define relations and properties of service orchestration. Orchestration is so far not part of the reference model, but if needed in future it should be implemented within the process model [15]. For modular process automation the orchestration of services is highly relevant, because services are executed in parallel and/or sequentially. Services will have relations to each other as well, which can be pointed out by a simple example: If a filling should be operated directly out of a tank, but a mixing should be fulfilled first, then an orchestration of the services "filling" and "mixing" is required.

c) Real world effects can include three different elements: Requested information, a change to a state of defined entities, or a combination of both [15]. Service consumers invoke a service for one of these three reasons. In the context of modular process automation, services for displaying values can be imagined which represent the outcome of the first mentioned real world effect. As mentioned in [5], services can be implemented by state-based control. This obviously fits ideally to the second real world effect.

## C. *SOCRADES*

The project *Service-oriented cross-layer infrastructure for distributed smart embedded devices* (SOCRADES) was an European research and development project addressing the SOA-based manufacturing paradigm (www.socrades.net) [19]. Basis for the SOCRADES-approach is the use of web services. Within SOCRADES, four topics were focused:

*a) Service-oriented architecture*

*b) Wireless sensor/actuator network*

*c) Enterprise integration*

*d) System engineering and management*

Within the SOA framework of SOCRADES, the system intelligence is implemented by physical agents that are embedded in smart devices (Fig. 4). These devices provide web services as an interface to communicate which each other [20]. All units are cooperating at the same hierarchical level; a higher process control level is not required. The autonomous units operate cooperatively, each is intelligent and proactive. Above this device-level the business control level is directly connected to the network of the units, also via web services. In order to compose all web-services, an orchestration unit, which provides orchestration web-services, is directly connected to the network. The functionality of orchestration could be based on a Petri net model of all autonomous units and their services. In the context of modular process automation, such an infrastructure would result in a different structure of the PCS and also of the communication between PCS and modules. Today's PCS-functions would be separated into different services as discussed in [21]. A superior control system would no longer be part of the infrastructure; instead separated web services would fulfill the PCSs functions.

b) The above-mentioned change in communication is defined in [20] as "… a network of equal participants without

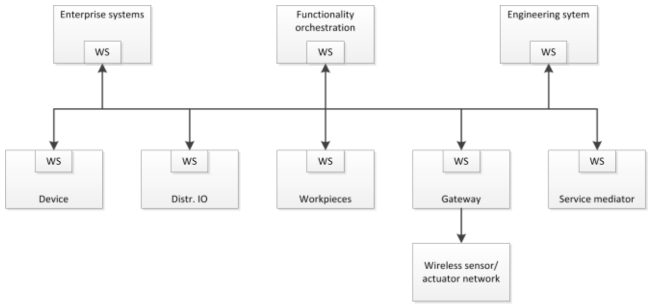


Fig. 4. SOCRADES cross-layer approach referring to [19]

any central control …". Therefore new communication protocols need to be specified. Future networks are seen as wireless networks between actuators and sensors [20]. In the context of modular process automation directly connected actuators and sensors are not required, but the concept can be used to build up a network between autonomous modules.

c) The main goal of SOCRADES is to bring device level functionality seamlessly and directly to levels of business management, without changing business application code. Therefore a definition for new integration concepts is required, which is described in [22]. A trend of usage of real time information also in business process leads to new enterprise integration approaches, such as new monitoring, effectiveness and maintenance analysis and optimizations [20]. In modular process automation communication between modules and PCS is still required. In order to use data from the module in business process levels the PCS will forward the data.

d) The fourth topic of SOCRADES addresses new tools and methods for system engineering and management to facilitate the system behavior. Without a higher control unit or system, the impact of the system is getting more complex. Therefore, new tools and methods are needed to achieve advantages of the new infrastructure without putting more effort into the control of the system [20]. Also for modular process automation, tools for module engineering as well as integration and control are required.

### *D. Process control with high-level Petri nets*

In addition to the more general SOA approaches described in sections *B.* and *C.* of this chapter, two approaches that are more specific will be described here. In [23] an approach of using high-level Petri nets to implement a service-oriented process control is proposed, which was introduced in the factory automation domain. Therefore, a classification of functional control components is proposed in [23]. *Mechatronic components* represent mechatronic devices, which fulfill basic functionalities. By combining the devices, more complex systems can be built. *Smart mechatronic components* differ from mechatronic components by an embedded smart control. Furthermore, they can participate in collaborative activities. *Process control components* provide services for process control. They are physically separated from the mechatronic devices and offer orchestration and composition of services, aiming at more complex functions. *Intelligence support components* provide decision services, aiming for support in flexibility and conflict resolutions [23]. According to the classification of components, a classification of control levels is given as well. At low level, the local control operates inside of mechatronic components. One level higher, a collaboration control is performed to combine devices to more complex functionalities. Aggregated control via specific control services within the process control components creates new services, based on basic services of mechatronic components.

In the context of modular process automation, smart mechatronic components could be understood as intelligent modules. Collaborative activities of different modules are so far not discussed within the process industries. A process control level is always required, but separation of control functionalities into services and into process control and intelligence support can also be seen in [21]. This hierarchical structure resembles the structure within ISA 106 [6].

To set relations between services, the approach of [23] uses the well-known mathematical foundation of Petri nets. Different types of relations are described by using Petri nets:

- Parallelism
- Concurrency
- Synchronization
- Memorizing
- Monitoring
- Supervising
- Resource sharing

In order to follow the OASIS reference model, the description of service relations should be defined in a process model using high-level Petri nets as a modelling language. Process control components are required to interpret the model and transform it into PLC-interpretable code. Transitions can be assigned to external input and output events from other entities or a higher level component via component-ports (Fig. 5). Active states of the Petri net model can be communicated to higher levels to interpret the state of the plant. The feature of composition and decomposition of Petri nets allows defining sub nets for each element of the main Petri net. The possibilities to structure the code hierarchically by composition lead to a simplified engineering of Petri nets [23].

The approach defines a way of communication via ports in different control levels, between different classes of components. Ports have to fit the requirements of SOA communication as stated in the beginning of this chapter.

Within a component the control is implemented by high-level Petri nets which provide the opportunity to define relations. By each Petri net of a control component the relations within the component are described.

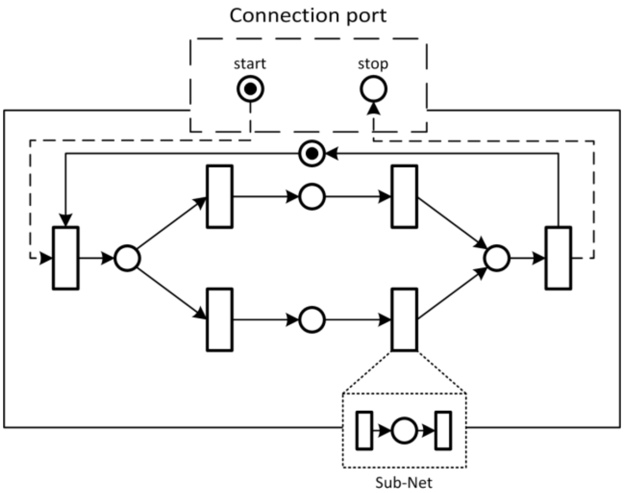


Fig. 5. High level Petri net structure of an example component referring to [23]

### *E. Implementation of service oriented architecture with function blocks*

In [24] an approach to implement SOA by the usage of function blocks according to [25] is introduced. Services are seen as encapsulated atomic elements that can be invoked by other services or by request-response messaging. Furthermore it is required to access process data such as actuator- and sensor-signals. The access is also implemented by using services. Interconnectivity between all involved devices is required, but services do not need to be allocated to particular hardware. Function block types according to [25] are considered as service types. Connections between function blocks represent messaging, thus, the communication between services. Communication is separated into message types and parameters, whereby messages are represented by event connections and parameters by data connections as described in the standard. The introduced approach origins in factory automation and decomposes manufacturing functionalities to the very basic level, e.g. "push workpiece". Each basic function is implemented as an atomic service. To build more complex functionalities, atomic services can be combined and orchestrated by the usage of a central coordination entity or by autonomous service choreography [24]. In addition, a combination of both coordination methods is allowed. A higher control level entity can invoke a service by messaging and the invoked service itself invokes subservices in a specific order. The orchestration of services is specific for the configuration of services implemented in the component and is executed by a specific state machine in the orchestration entity. It is possible to use function blocks of [25] to describe the orchestration and relations of services in a proprietary solution per configuration of devices. However, neither the state machine nor the orchestration function block is extensible for new devices.

In [24] the lack of a standard form for representing service relations is stated. In context of modular process automation this lack is also present, but a generic description model is needed to obtain flexibility of modularization.

## IV. Criteria of Modular Process Automation

The following chapter defines criteria of requirements which need to be covered by a SOA approach in order to be suitable for modular process automation. Only criteria regarding control and communication in the context of modular process automation are considered. The criteria are deduced from requirements mentioned in chapter II and general SOA requirements of chapter III-A.

General requirements as the typical functionalities of a PCS and requirements regarding testing, verification, piloting and monitoring of modules do not affect the choice of a communication and control infrastructure.–Therefore, these requirements are not mentioned specifically below.

a) *Availability of superior control system* is required to control integrated modules and included services. A control by cooperation between autonomous modules is most probably not possible due to the absence of discrete workpieces. In addition, a control of overall interlocks is required. The superior control system can be implemented by a PCS. All functionalities of a "classical" PCS, as mentioned in chapter II are required, but can be partly shifted to modules.

b) *Implementation of intelligent modules* is required. Each module is automated using a PLC including all accessible services of the devices within the module.

c) *Choice of free granularity of services* is important to achieve acceptance within the domain of module manufacturing. Dissociation from competition by using different detailed services and modules will most probably lead to a rich supply of modules within the process industries. Additionally, module integrators will be offered higher flexibility in plant design.

d) *Service control via state-machine* is required by [5] as the leading organization of PCS-suppliers in Germany. In particular, the control should build upon the state-machine of ISA 88 [7].

e) *Service-response about active state* in order to be able to control the process is required.

f) *Modelling of service-relations* is required in order to assure functional safety. As seen in introduced approaches, relations between services are discussed in the literature. To avoid precarious combinations of active states of different services, a model of all service-relations is required.

## V. Analysis

In the following, all SOA approaches discussed in Chapter III are analyzed regarding the requirements stated in Chapter IV. Therefore, three ratings are introduced:

- ✓ *Requirement is completely fulfilled*
- (✓) *Approach bears a possibiltiy to fulfill the requirement*
- o *Requirement is not fulfilled*

TABLE I. ANALYSIS IF OASIS AND SOCRADES APPROACHES FULFILL THE CRITERIA OF MODULAR PROCESS AUTOMATION

| Criteria | OASIS reference model | SOCRADES |
|---|---|---|
| a) Availability of superior control system | ✓ | o |
| b) Implementation of intelligent modules | (✓) | ✓ |
| c) Choice of free granularity of services | ✓ | ✓ |
| d) Service control via state-machine | (✓) | (✓) |
| e) Service-response about active state | ✓ | (✓) |
| f) Modelling of service-relations | (✓) | ✓ |

The OASIS reference model fulfills all requirements completely or provides at least possibilities for fulfillment (TABLE I.). In the model, intelligent modules are not described, but the infrastructure allows an implementation of autonomous modules with its own PLC. Although a control via service-states is not described in detail, within the service description and by communication of the correct control an implementation is possible Within the OASIS reference model service-relations are not specified, but a possible storage for them is defined. If service-relations are required within an implementation based on the reference model, all relations should be stored within the process model.

The SOCRADES approach does not fulfill all requirements of modular process automation (TABLE I.). Because of the origin of the approach in the domain of factory automation, a central control system is not foreseen in SOCRADES. The approach defines ways of interaction and cooperation of involved modules based on order information from business management system alone. All devices or modules of the approach bring along their own intelligence, and by combination of single devices the granularity of services is variable. State-machines are not mentioned in the approach, but an implementation by Petri nets for orchestration services. State-machines can be translated into Petri nets, so a service control via states seems possible. Because of the absence of a superior system, also the response of the currently active state is not part of the approach, but the infrastructure allows bidirectional communication. As mentioned before, service relations are part of the orchestration service.

The implementation approach of SOA by high-level Petri nets fulfills nearly all stated requirements completely (TABLE II.). A superior control system can be added to this approach but is not defined. Via ports of process control components, communication to a higher level is assured. Intelligent modules are represented within the approach by smart mechatronic components or even by process control components, depending on the complexity of the service, which also fulfills the requirement of free granularity of services. Control of services is implemented by high-level Petri nets, which can describe state-machines. If a superior control system is added to the approach, a response of the active state of each Petri net can be communicated via ports of control components. As mentioned in the SOCRADES approach Petri nets are suitable to describe service relations.

TABLE II. ANALYSIS IF HIGH-LEVEL PETRI NET AND FUNCTION BLOCK APPROACHES FULFILL THE CRITERIA OF MODULAR PROCESS AUTOMATION

| Criteria | High-level Petri nets according to [23] | Function blocks according to [24] |
|---|---|---|
| a) Availability of superior control system | (✓) | (✓) |
| b) Implementation of intelligent modules | ✓ | (✓) |
| c) Choice of free granularity of services | ✓ | ✓ |
| d) Service control via state-machine | ✓ | ✓ |
| e) Service-response about active state | ✓ | ✓ |
| f) Modelling of service-relations | ✓ | ✓ |

The approach to use IEC 61499 function blocks to implement SOA fulfills all stated requirements or can be adapted to do so. A superior control system is not mentioned in the approach, but is a possible extension, as the standard is a progression of well-known function blocks of [26] within PLCs, which communicate with PCSs. The standard also allows implementing intelligent modules using function blocks. The infrastructure of the approach gives free choice of service-granularity. Thereby each service is controlled via its own state-machine, which is also defined within [25]. Based on the in- and outputs of each function block, a response to a superior control system, if installed, is assured. As stated in [24], service relations are highly important to run a process industry plant safely. By using function blocks all relations can be described for a specific configuration. However an overall relationship-model is not given.

To summarize, the analysis of already introduced SOA approaches shows that the concept of SOA is applicable to the domain of modular process automation. The desired encapsulation of process functions in services of different granularity is possible. Each service can be accessed and invoked by a control component, which can be an orchestration service or – as required in modular process automation – an adaption of a PCS.

## VI. CONCLUSION AND FUTURE WORK

Within this contribution, different SOA approaches have been discussed and analyzed with respect to the requirements of modular process automation. It has been shown that SOA, as an infrastructure paradigm for communication and state-based control, fulfills the requirements of modular process automation. All SOA approaches discussed here, except of one, fulfill the six derived requirements completely or provide a possibility for an extension of the approach by which a fulfillment can be achieved. This result confirms SOA as a suitable paradigm for communication within modular process automation and shows that appropriate approaches are already available for the use in the process industries. Especially service orchestration by usage of high-level Petri nets seems promising, as Petri nets are well known in the domain of automation and are used in many different models. Also the possibility to describe the requested state-machines by the same description language as service relations is an advantage.

The future work of the authors will focus on the development of a SOA approach for modular process automation according to the OASIS reference model and based on already available approaches to fit all requirements completely. Therefore, a general state-machine for services will be developed, based on existing standards like [7]. Furthermore a meta-model of service relations will be developed which shows how services influence each other.

## REFERENCES


[1] S. Lier and M. Grünewald, “Net Present Value Analysis of Modular Chemical Production Plants”, Chem. Eng. Technol, vol. 34, no. 5, pp. 809–816, 2011.

[2] T. Holm, M. Obst, A. Fay, L. Urbas, U. Hempen, T. Albers, and S. Kreft, “Engineering method for the integration of modules into fast evolving production systems in the process industry”, in 2015 IEEE International Conference on Automation Science and Engineering (CASE), pp. 1042–1047.

[3] J. Rottke, F. Grote, H. Fröhlich, D. Köster, and J. Strube, “Efficient Engineering by Modularization into Package Units”, Chemie Ingenieur Technik, pp. n/a-n/a, 2012.

[4] F3 Factory. Available: http://www.f3factory.com/scripts/pages/en/home.php.

[5] ZVEI, “White Paper: Module-Based Production in the Process Industry – Effects on Automation in the “Industrie 4.0” Environment: Recommendations of the Modular Automation Working Group Following Namur Recommendation NE 148”, Frankfurt am Main, Feb. 2015.

[6] Procedure Autmation for Continous Process Operations, ISA-TR 106.00.01.

[7] Batch Control Part 1: Models and Terminology, ANSI/ISA 88.00.01.

[8] Y. V. Natis, “Service-Oriented Architecture Scenario”, Available: http://info.lnpu.edu.cn/website/jpkc/rjgc/ywzl/Service-Oriented%20Architecture%20Scenario.pdf (2015, Oct. 14).

[9] L. Evertz, “Laying a basis for service systems in process control”, in 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), pp. 1–8.

[10] M. Bichler and Kwei-Jay Lin, “Service-Oriented Computing”, Computer, vol. 39, no. 3, pp. 99–101, 2006.

[11] C. Bramsiepe, S. Sievers, T. Seifert, G. D. Stefanidis, D. G. Vlachos, H. Schnitzer, B. Muster, C. Brunner, J. Sanders, M. E. Bruins, and G. Schembecker, “Low-cost small scale processing technologies for production applications in various environments—Mass produced factories”, Chemical Engineering and Processing: Process Intensification, vol. 51, pp. 32–52, 2012.

[12] Automation Requirements relating to Modularisation of Process Plants, NE 148, 2013.

[13] T. Tauchnitz, “Die “neuen Prozeßleitsysteme - wohing geht die Reise?”, atp - Automatisierungstechnische Praxis, vol. 38, no. 11, pp. 12–23, 1996.

[14] M. Obst, T. Holm, L. Urbas, A. Fay, S. Kreft, U. Hempen, and T. Albers, “Semantic description of process modules”, in 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), pp. 1–8.

[15] C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, and R. Metz, “Reference Model for Service Oriented Architecture 1.0.”, Available: http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html (2015, Oct. 15).

[16] M. Schlütter, U. Epple, and T. Edelmann, “On Service-Orientation as a New Approach for Automation Environments”, in ARGESIM report, vol. 35, Proceedings / MATHMOD 09, I. Troch, Ed, Vienna: ARGESIM Publ. House, 2009, pp. 2426–2431.

[17] D. Henneke, M. Elattar, and J. Jasperneite, “Communication Patterns for Cyber-Physical Systems”, in 20 th IEEE International Conference on Emerging Technologies and Factory Automation, 2015.

[18] H. Haas and A. Brown, “Web service Glossary”, Available: https://www.w3.org/TR/ws-gloss/ (2016, Mar. 06).

[19] European IST FP6 project, SOCRADES - Service-Oriented Cross-layer Infrastructure for Distributed smart Embedded devices. Available: www.socrades.net.

[20] A. Cannata, M. Gerosa, and M. Taisch, “SOCRADES: A framework for developing intelligent systems in manufacturing”, in 2008 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp. 1904–1908.

[21] S. Karnouskos, A. W. Colombo, T. Bangemann, K. Manninen, R. Camp, M. Tilly, M. Sikora, F. Jammes, J. Delsing, J. Eliasson, P. Nappey, J. Hu, and M. Graf, “The IMC-AESOP Architecture for Cloud-Based Industrial Cyber-Physical Systems”, in Industrial Cloud-Based Cyber-Physical Systems: The IMC-AESOP Approach, A. W. Colombo, T. Bangemann, S. Karnouskos, J. Delsing, P. Stluka, R. Harrison, F. Jammes, and J. L. M. Lastra, Eds, s.l.: Springer International Publishing, 2014, pp. 49–88.

[22] S. Karnouskos, O. Baecker, L. M. S. de Souza, and P. Spiess, “Integration of SOA-ready networked embedded devices in enterprise systems via a cross-layered web service infrastructure”, in 2007 IEEE Conference on Emerging Technologies & Factory Automation (EFTA 2007), pp. 293–300.

[23] J. M. Mendes, P. Leitao, A. W. Colombo, and F. Restivo, “Service-oriented process control using High-Level Petri Nets”, in 2008 6th IEEE International Conference on Industrial Informatics (INDIN), pp. 750–755.

[24] W. W. Dai, J. H. Christensen, V. Vyatkin, and V. Dubinin, “Function block implementation of service oriented architecture: Case study”, in 2014 12th IEEE International Conference on Industrial Informatics (INDIN), pp. 112–117.

[25] Function blocks - Part 1: Architecture, IEC 61499-1, 2012.

[26] Programmable controllers - Part 3: Programming languages, IEC 61131-3, 2013.