An Engineering Approach for industrial SoA-based Systems of Systems

Kevin Nagorny

Department of Technology
University of Applied Sciences Emden/Leer (HSEL)
26723 Emden, Germany
Kevin.Nagorny@technik-emden.de

Elena Freijo

Department of Technology
University of Applied Sciences Emden/Leer (HSEL)
26723 Emden, Germany
elena.freijo@gmail.com

Armando Walter Colombo

Department of Technology
University of Applied Sciences Emden/Leer (HSEL)
26723 Emden, Germany
awcolombo@technik-emden.de

Jerker Delsing

Lulea University of Technology
97187 Lulea, Sweden
jerker.delsing@ltu.se

Abstract— Current implemented manufacturing and continuous production process systems are strongly being influenced by the fusion of mechatronics, communication, control and information technologies. The miniaturization of control and automation devices with embedded intelligence is transforming the production systems into a very large infrastructure, where different systems interact in a structural and behavioral manner for pursuing common goals. The result is a very large complex system-of systems (SoS). The application of the service-oriented architecture (SoA) paradigm is appearing as a promising approach to develop and implement the control and management of those SoS. This paper describes an approach to support the engineering of service-oriented based Industrial System-of-Systems and presents the first results of application of the approach to a real heterogeneous system composed of a district-heating, transportation and an electrical power distribution system.

Keywords— *Service-oriented architecture, System-of-Systems, Service Orchestration, SoA-based control, SoA-based Engineering*

I. INTRODUCTION

This paper describes an approach to support the engineering of industrial systems-of-systems [19][24] based on the application of the SoA paradigm [1] and model-based engineering methods [32].

A SoA infrastructure implemented in a highly distributed system, like a traditional district-heating system or a transportation system generates new possibilities to put those completely different systems together, i.e., facilitating the connectability and interoperability of their heterogeneous components [8][11][13][16][17]. Completely different systems from physical point of view are virtualized by the Service-orientation and linked in a Cyber-level by one common web service based communication network [18][21].

Under this innovative paradigm change in the system engineering domain, a unified network technology provides the reachability of smart devices and systems or system's

components [6][7] in a cloud of services (service-cloud) as shown in Figure 1 [31]. The service-cloud represented in the upper part of Figure 1 includes different kind of services, which are embedded in smart automation devices able to work as client and/or servers in the information communication control technology (ICCT) system.

Figure 1: Service-oriented Integration of heterogeneous ICCT-Systems

For example, a service of a district-heating system could be “heating-valve-1 close”. As soon as the SoA-based heating system is setup, this and all other services will immediately be registered and exposed in the network. A heating process associated to the energy consumption of a house will need e.g. this 1 service in order to reduce the energy consumption [26]. The system which is managing the energy will now find the exposed “heating-valve-1 close” Service exposed in the SoA-based network. In a similar way, any other system integrated in the same SoA-network will be able to access the same service and to use it [22], e.g., in a direct way or composing, orchestrating it for generating new emergent functionalities (services) [30][29][28].

1. The system is "evolvable", in the way that each time a component system connects to or disconnects from the network, the functionalities of this component exposed as "services" are registered and appear ready for being used by other components of the network or disappear from the list of ready-registered services [27][22].

3. The “Service-Bus” addressed in Figure 1 is another form of representing the “Service-Cloud”. This Cyber-representation of the System-of-Systems is completely independent of the geographical position of the Physical component generating the “Services” [13].

II. AN ENGINEERING APPROACH FOR SOA-BASED SYSTEMS-OF-SYSTEMS

The goal here is to create functional topologies inside each system and make them work together. This means to combine the available services in a way that will allow the desired operation of the systems and to find a connection between them. As pointed out below, this method is tailored to the SoA approach since the topologies can be easily modified and adapted to new requirements enabling flexibility, interoperability. The idea is an easy developable Systems-of-Systems engineering approach for the SoA-paradigm.

This method follows a “top-down” philosophy, the technical details or improving efficiency are among others issues that will be addressed later. Each system can be represented in the form of a black box. Until this point it doesn’t matter how the system works internally.

Figure 2 is an introduction in the used symbols in the high level engineering process. These symbols will support a “drag-and-drop” engineering method. Note: Although the contents of the rows 3, 4, 5 and 6 appear having same levels, the semantic contents are completely different. This is clearly shown in the engineering tool using icons with different colors.

Blackboxes (Legend)

	DHS	Electrical Outlet	iPark	Car HMI	iRoad
Service (S)
Blackbox (BB)
- Input
- Output
Event Service Arrow
Orchestrator (Orch)

The columns in Figure 1 define different domains in an example of a smart environment in the area of building automation. Domains are clustered systems which are offering a whole function of one conventional system as services. One conventional independent system/domain in a building is the heating system. Another domain is the electrical system/domain; etc. Generally are the symbols of each domain shown in a different color to simplify the distinguishing between different domains during the engineering process.

In the rows of Figure 2 are shown the different symbols of the used elements. They will be described in sequence row by row.

The symbol of a service is shown in the first row. It includes always the “S” (“=”Service”).

The symbol of a BlackBox is a rectangle labeled with the name (identifier) of the system.

The interface of a BlackBox is shown by input- and output arrows. Input arrows are always red and labeled with the function of this arrow. Output arrows are always colored in blue. The label is equivalent with the input arrows.

The interface to an event-based service is shown with an orange arrow. A more detailed picture of this symbol is also shown in Figure 3. An event-based service needs a subscriber which subscribes the service. “Event” means that the subscriber will be notified, if a subscribed event appears. An example is the event: “the temperature reached 30 °C”.

Figure 3: Representation of an event based service

The arrow is connected to the event trigger and the circle to the subscriber. An event interface of a BlackBox is a circle or an arrow, depending on the position of the Event Trigger/Subscriber.

The symbol of an orchestrator is a hexagon. . It is always labeled with the shortcut “Orch” (“=”Orchestrator”).

B. Domain- and system categorization

In this method a system which is represented as BlackBox, is a set of services, orchestrators and other BlackBoxes/systems. Each new sub-system is different in characteristics, properties or functions.

Figure 4: ISA-95 Pattern

As a starting point a categorization of domains is necessary to simplify the following engineering process. This part could be seen as an ISA`95pattern [9][10] (see Figure 4). Domains are clustered systems which are offering the functions of a whole conventional system. Inside of each domain are services on which are providing functions of different layers of the ISA`95 levels. To identifier a domain, it is helpful to analyze a conventional system in the same area.

To support the visual abstraction, it is useful to select a different color for each domain (see Figure 5).

Figure 5: Representation of systems as black boxes

Note: Although the domains are categorized and bundled in BlackBoxes, domains are in the further engineering process interoperable. An emergent an evolvable property is anytime given, such as a cross-domain- and cross-layer communication.

This method is a “Top-down” approach. Only the highest abstraction of a system is a domain. All BlackBoxes inside of a domain are called “systems”. The definition procedure of the systems is equivalent with the domain definition.

The result after this procedure should be system-of-systems as shown in Figure 6.

Figure 6: Representation of sub-systems

After the domain analyzation process, the interfaces have to be defined. The question is, which inputs (e.g. control; setting parameter) and which outputs (e.g. monitoring) are needed from each domain.

C. Interface definition

Once a system is known, the interface has to be defined. It has to be defined, which services should be provided from the system.

There are inputs and outputs for normal services and event-based services. Input services are e.g. SET services (e.g. SET “actuator1”). Output services are GET services (e.g. GET “sensor1”). The engineer has to use the even-based interface if an actuator has to be set caused of an event (event-SET-input) or if information is needed caused of an event (event-GET-output).

From the graphical point of view the input arrows have to be on the left side and the output arrows of the right side.

One example BlackBox with interface is shown in Figure 7.

Figure 7: Representation of a system with its interface

D. Service- and orchestrator integration

After the definition procedure of all domains, systems and interfaces; the next step is the service- and orchestrator integration. This step is a bottom-up procedure.

The smallest system is a BlackBox with one included service (as shown in Figure 8) or one included orchestrator (as shown in Figure 9). The needed service-functions are shown as interface of the BlackBox. In the example of Figure 8 is shown a Valve service included in a BlackBox. The Valve service provides two service-functions. The first service-function is “PUT Position” which is able to open or close the Valve. The second service-function is “GET Position” which is able to return the current position of the Valve. The “GET Position” service-function is also available as event-based Service.

Figure 8: Example smallest system (Service)

The procedure for the orchestrator integration is analog to the service integration (see Fig. 9). An Orchestrator is a service which is able to orchestrate two or more Services/Orchestrators. An Orchestrator realizes an abstraction of control-functions for which are more Services/Orchestrators necessary or is able to realize synchronization, a control loop or a regulation. An Orchestrator is optional a Service provider and is necessary a Service requester.

Figure 9 shows an example. The BlackBox “Transport Belt Synchronizer” includes an Orchestrator which is able to control two different transport belts. Its task is to set two belts parallel into the same belt speed. The Orchestrator controls both belts over the “SET” outputs and is itself reachable over a Service “SET Belt Speed”. A Service requester can use this Service to synchronize the belt speed of both belts by using only on Service. A further abstraction by using one Orchestrator as shown in Figure 8 to synchronize two further Orchestrators and as result four transport belts is anytime possible.

Figure 9: Example smallest system (Orchestrator)

E. Topology generation

After the definition procedure of all domains, systems and interfaces; the next step is the topology generation. That means that all systems inside of domains have to be connected by using the interfaces.

Three rules are given.

1. An output of a system is connectable with an input of a system.
2. An input of a system is connectable with an output of a system.
3. An event-based interface with the subscriber side (circle) is connectable with and event-based interface with the event trigger side (arrow).

The logic of a system will be validated over a Web Services Description Language (WSDL) matching.

Figure 10: Topology generation

The connection of all systems has to be made bottom-up. The final step is the cross-domain communication between all domains. Figure 10 is showing an example of the topology generation. One services with identic color means that they are from the same domain. Connections between services in the same domain are connections on the same layer or cross-layer connections (cross-layer composing). Connections between services in different domains are cross-domain connections (cross-domain composing).

III. PROTOTYPE IMPLEMENTATION

The engineering method is being tested in a prototype environment as shown in Figure 11.

Figure 11: Heterogeneous Cross-Domain ICCT-System of Systems.

The environment consist of a Cross-Domain Application System integrating heterogeneous ICCT-Systems:

- District heating substation system [2][3][4][5]
- Home parking system [36]
- Electrical Outlet System
- Car HMI system [32]
- iRoad system (is able to exchange information to the Car HMI system) [33][34][35][37]

For the addressed domains were developed different cyber-physical systems which are able to expose the different functions as “Services”.

Figure 12: Modeling the topologies with a drag and drop engineering method.

The engineering approach has been validated using the story scenario described below:

- Default operation of District Heating (DH) system
- External electrical outlets turned of
- Monitoring incoming cars to house
- Detecting incoming car
- Determining if car should be parked outside or inside house based on outdoor temperature

- Information to the driver
- If suggestion to park inside open the garage port
- If low temperature and parking full enable electrical outlet for car heating
- If parked inside inform DH system of addition heat source
- Tune space heating such that additional heat source is used in a favorable way
- System level information - ΔT
- DH control loop tuning, changes of set points.

Following the engineering process described in the chapters before, it has been possible to generate the full topologies of the scenario. Figure 12 shows the highest abstracted view of it, highlighting the different domains and the interfaces of some of the component systems.

IV. CONCLUSION AND OUTLOOKS

This paper described an approach to support the engineering of industrial systems-of-systems based on the application of the SoA-paradigm and formal engineering methods.

As in chapter 3 described, a SoA-infrastructure for conventional systems [25][20][23][24] like e.g. in the production or building automation domains generates new possibilities to put structurally and dynamical (behavioral) completely different systems, e.g. district heating and electrical system, together. From the virtualization of the completely different systems (from the physical point of view) in a SoA-infrastructure results a Cyber-environment [14][15] with a common web service based communication and information network.

All virtualized systems expose their functionalities as services, which are reachable in a kind of “service-cloud”. The described engineering method provides and easy usable system-of-systems tool which supports emergent and evolvable properties provides a drag and drop engineering path for complex systems of systems [26].

Future works are related to the development of a supporting engineering method by using e.g., formal method and the development of a data model.

A formal method is necessary for the control of complex control structures. Recent work in this area are control structures by using high-level petri nets for the control, monitoring, energy management, etc. for smart devices [32]. A data model is necessary for the communication between different systems. Further work has to deal with the question, which information is necessary for the next system, which information is optional, etc. A full semantic between all systems has to be defined. Finally, a model driven configuration of all virtualized components has to be developed and implemented. If services and orchestrators are providing security possibilities (certificates, MAC filter, passwords, etc.), it is possible to configure services and orchestrators automatically for an exclusive access for the in the engineering tool modeled connections.

ACKNOWLEDGMENT

The authors would like to thank the partners of European Commission funded projects IMC-AESOP (www.imc-aesop.eu) and the members of the I2AR Institute (www.i2ar.de) for the fruitful discussions.

REFERENCES

- [1] Colombo, A. W.: SOCRADES: Steps Towards the Factory of the Future Projects Magazine EU, British Publishers Inc. 12, 20-23. (<http://viewer.zmags.com/publication/4c086127#/4c086127/1>)
- [2] Gustafsson, J.; Mäkitaavola, H.; Delsing, J. & van Deventer, J. Integration of an ip based low-power sensor network in district heating substations The 12th International Symposium on District Heating and Cooling, 2010
- [3] Gustafsson, J.; Delsing, J. & van Deventer, J. Validation of a district heating substation model using a wireless sensor network approach Proc. Int. District heating Symposium, Orlando, 2008, 8
- [4] Gustafsson, J.; Delsing, J. & van Deventer, J. Improved district heating substation efficiency with a new control strategy Applied energy, Elsevier, 2010, 87, 1996-2004
- [5] van Deventer, J.; Gustafsson, J. & Delsing, J. Controlling district heating load through prices Systems Conference (SysCon), 2011 IEEE International, 2011, 461-465
- [6] Garg, D.P.; Poppe, C.D.; , Coordinated robots in a flexible manufacturing work cell, Advanced Intelligent Mechatronics, 2001. Proceedings. 2001 IEEE/ASME International Conference on , vol.1.
- [7] Bing Liu; Siew-Hwee Choo; Shee-Ling Lok; Sing-Meng Leong; Soo-Chee Lee; Foong-Ping Poon; Hwee-Har Tan; , Finding the shortest route using cases, knowledge, and Djikstra's algorithm, IEEE Expert , vol.9, no.5, pp.7-11, Oct 1994.
- [8] Cichocki, A.: Workflow and process automation: concepts and technology Kluwer international series in engineering and computer science; Kluwer Academic Publishers (1998).
- [9] THE WBF BOOK SERIES-APPLYING ISA 95. Implementation Experiences. Wbf Book Series, World Batch Forum, Wbf, Momentum Press, 2010.
- [10] Dennis Brandl & Peter Owen: A Tutorial on the ANSI/ISA95 Enterprise/Control BR&L Consulting. www.brlconsulting.com/.../2003-09%20IEE%20... 6 Jan 2001 – Dennis Brandl Peter Owen. BR&L Consulting Eli Lilly & Co. 2.
- [11] Collaborative Manufacturing Management Strategies. ARC White Paper. Gorbach and Mick 2002. ARC Advisory Group.
- [12] Collaborative Automation: The Platform for Operational Excellence. ARC White Paper. Polsonetti 2003. ARC Advisory Group.
- [13] Cândido, G.; A. W. Colombo; J. Barata and F. Jammes: Service-Oriented Infrastructure to Support the Deployment of Evolvable Production Systems. IEEE Transaction on Industrial Informatics, Vol. 7 Num. 4, November 2011, pp. 759-767.
- [14] Mendes, M. J.; J. M.; P. Leitão; F. Restivo, A. W. Colombo and A. Bepperling: Engineering of service-oriented automation systems: a survey. International Journal of Digital Manufacturing, Vol. 1, N. 1, pp. 33-38, 2008.
- [15] Taisch, M.; A. W. Colombo; S. Karnouskos and A. Cannata: SOCRADES Roadmap: An Executive Summary. M. Taisch and A. W. Colombo (Eds.). EU Press. 2009.
- [16] Colombo, A. W. and S. Karnouskos: "Towards the Factory of the Future: A Service-Oriented Cross-Layer Infrastructure". The European Telecommunications Standards Institute (ETSI) Book on: "ICT Shaping The World - A Scientific View", chapter 6. John Wiley and Sons Ed., April 2009.
- [17] Jens Eliasson, J.; R. Kyusakov; J. Delsing; J. Nessaether; A. W. Colombo; F. Jammes; S. Karnouskos and C. Diedrich: A Migration Approach to a SOA-based Architecture for Next Generation Process Control and Monitoring. Proc. of the 37th IEEE International Annual Conference on Industrial Electronics (IECON'11), Melbourne, Australia, 7-10 November 2011.
- [18] Zhang, L.; Guo, H.; Tao, F.; Luo, Y.L.; Si, N.; , "Flexible management of resource service composition in cloud manufacturing," Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on , vol., no., pp.2278-2282, 7-10 Dec. 2010.
- [19] Idoughi, M. Kerkar, and C. Kolski, Towards new web services based supervisory systems in complex industrial organizations: Basic principles and case study, Comput. Ind., vol. 61, pp. 235-249, April 2010. [Online].
- [20] Supervisory Control and Data Acquisition (SCADA) Systems. National Communications System (NCS), Technical Information Bulletin 04-1, Technical Report, October 2004. www.ncs.gov/.../tech_bulletins/2004/
- [21] Colombo, A. W. and R. Harrison: Modular and Collaborative Automation: Achieving Manufacturing Flexibility and Reconfigurability. Int. J. Manufacturing Technology and Management, Vol. 14, Nos. 3/4, pp. 249-265. INDERSCIENCE PUBLISHERS. 2008.
- [22] Harrison, R.; Colombo, A. W., West, A. A. and Lee S.M.: Reconfigurable Modular Automation Systems for Automotive Power-Train Manufacture, International Journal of Flexible Manufacturing Systems, Springer, 2006, 18:175-190, DOI 10.1007/s10696-006-9008-y.
- [23] V. Malhotra, T. Raj and A. Arora: Reconfigurable manufacturing system: an overview. International Journal of Machine Intelligence, Volume 1, Issue 2, 2009, pp- 38-46.
- [24] Reconfigurable Manufacturing Systems and Transformable Factories. Dashchenko, Anatoli I. (Ed.). Springer. 2006.
- [25] Changeable and Reconfigurable Manufacturing Systems. Springer Series in Advanced Manufacturing. ElMaraghy, Hoda A. (Ed.). 2009.
- [26] P. Kennedy, V. Bapat, and P. Kurchina, In Pursuit of the Perfect Plant. Evolved Technologist, 2008.
- [27] M. Jamshidi, Ed., Systems of Systems Engineering. plus 0.5em minus 0.4emCRC Press, Nov. 2008.
- [28] G. Lewis, E. Morris, P. Place, S. Simanta, D. Smith, L. Wrage, "Engineering Systems of Systems," in IEEE International Systems Conference (SySCon 2008), 7-10 April 2008.
- [29] S. Simanta, E. Morris, G. Lewis, D. Smith, "Engineering Lessons for Systems of Systems Learned from Service-Oriented Systems," in IEEE International Systems Conference (SySCon 2010), 5-8 April 2010.
- [30] Colombo, A. W.; S. Karnouskos and J. Delsing: Learning from SoA-based Principles to Engineering Large Industrial Systems of Systems. Notes of the Complex Systems Engineering to Systems-of-Systems Workshop, European Commission, Brussels, Belgium, July 2011
- [31] Karnouskos, S and A. W. Colombo: Architecting the next generation of service-based SCADA/DCS system of systems. Proc. of the 37th IEEE International Annual Conference on Industrial Electronics (IECON'11), Melbourne, Australia, 7-10 November 2011.
- [32] Kevin Nagorny, Armando Walter Colombo, Uwe Schmidtman, A service- and multi-agent-oriented manufacturing automation architecture: An IEC 62264 level 2 compliant implementation, Computers in Industry, Available online 27 August 2012, ISSN 0166-3615, 10.1016/j.compind.2012.08.003.
- [33] Birk, W.; Eliasson, J.; Lindgren, P.; Osipov, E. & Riliskis, L. Road Surface Networks technology enablers for enhanced ITS Vehicular Networking Conference (VNC), 2010 IEEE, 2010, 152-159
- [34] Birk, W.; Osipov, E. & Eliasson, J. iRoad Cooperative Road Infrastructure Systems for Driver Support Proceedings of the 16th ITS World Congress, 2009
- [35] Eliasson, J. & Birk, W. Towards road surface monitoring: Experiments and technical challenges Control Applications(CCA) & Intelligent Control,(ISIC), 2009 IEEE, 2009, 655-659
- [36] Eliasson, J.; Van Deventer, J. & Johanson, M. An ad-hoc bluetooth sensor network for automotive testing Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE, 2008, 179-180
- [37] Nordlander, J. & van Deventer, J. Reactive Programming of Event-Driven Automotive Systems 2005