Enabling model-based engineering of service-oriented Architectures within complex industrial Systems

Christoph Binder, Goran Lastro and Christian Neureiter
Josef Ressel Center for Dependable
System-of-Systems Engineering
Urstein Sued 1, A–5412 Puch/Salzburg, Austria
firstname.lastname@fh-salzburg.ac.at

Arndt Lüder
Otto-v.-Guericke University
Universitätsplatz 2
D-39106 Magdeburg, Germany
arndt.lueder@ovgu.de

***Abstract*—Caused by the transformation of traditional production lines towards ubiquitous interconnected manufacturing networks in the context of Industry 4.0, the original product-orientation is evolving towards service-oriented systems. Thus, various service-oriented architectures (SOAs) have been recently proposed in order to deal with the increasing complexity in such production systems. However, most of those reference architectures look good on paper, but are missing practical applications, as this is also the case with the Reference Architecture Model Industrie 4.0 (RAMI 4.0). Therefore, in order to increase the usability of RAMI 4.0 in terms of its service-orientation, a detailed architecture definition of its Communication Layer is proposed in this paper. The SOA thereby integrates the characteristics of the ISO 42010 and provides a particular domain-specific language (DSL). Finally, the applicability and usability of the resulting architecture is evaluated with the help of a real-world case study considering a manufacturer of copper-plated metal plates.**

***Index Terms*—System Architecture, Service-oriented Architecture (SoA), Industrial Internet of Things (IIoT), Reference Architecture Model Industry 4.0 (RAMI 4.0), Domain-specific Systems Engineering (DSSE)**

I. INTRODUCTION

Systems engineering of current and future industrial systems is constantly increasing in complexity. This is mainly attributed to new technologies originating from the outcomes of industrial applications or research projects. Those results support the manufacturing of products in many different ways throughout the whole life cycle. In more detail, ubiquitous interconnection leads to the automation of repetitive or manual tasks and additionally supports information exchange at the same time. Better known by the terms “Industry 4.0” or Industrial Internet of Things (IIoT), system elements like Cyber-physical Systems (CPS) support this transformation towards a resource-efficient and optimized production of individual products. This behavior leads to the generation of new business models for manufacturing companies. One major aspect dealing as an example for such a change resulting from this transformation is the drifting away from original product-centric manufacturing towards service-orientation [1]. In traditional production lines, raw materials are processed into finished products, which are then sold to potential customers. As far as the manufacturer is concerned, no further obligations need to be adhered to. However, regarding IIoT-based and service-oriented aspects, the interconnection between manufacturers as well as customers reaches over the entire life-cycle of manufactured products. Thus, before actually creating the product, customers could individually configure it according to their desires. On the other hand, after selling the product, additional services could be offered. This leads to increasing complexity in contemporary manufacturing systems.

In order to deal with this increasing complexity in service-oriented manufacturing systems, the importance of Service-oriented Architectures (SOAs) is becoming more and more obvious in recent years [2]. Additionally, Model-based Systems Engineering (MBSE) appears to be a suitable method when it comes to develop such complex systems. With this method, different perspectives can be generated to support mutual systems engineering by considering the design principles “separation of concerns” as well as “divide and conquer”. This allows to regard all aspects of such interwoven structures of System of Systems (SoS), as manufacturing systems are. In addition, the components providing different services can be place within the architecture according to their characteristics. Thus, in order to provide a methodology integrating the aforementioned concepts while engineering complex industrial systems, the Reference Architecture Model Industrie 4.0 (RAMI 4.0) has been proposed. This three-dimensional reference architecture provides axes as well as layers to locate Industry 4.0 components with all functions and services. Doing so, RAMI 4.0 itself is constituted based on a SOA. Therefore, based on this reference architecture, the technical realization of Services should be organized by utilizing a service-model as part of the whole system architecture [3]. By exemplary applying the basic service of Open Platform Communications Unified Architecture (OPC UA), Industry 4.0-compliant communication can be ensured within the manufacturing system. In consequence, the Communication Layer of RAMI 4.0 is especially aligned to deal with service-oriented communication in those systems, as interfaces of the components as well as the availability of their services should be located at this level.

However, beyond the official standardized definition, there are almost none examples how to make use of the Communication Layer in order to describe the communication infrastructure of industrial systems. This may be caused by insufficient specifications within the standard itself, as pointed out in [4]. Nevertheless, in order to ensure the applicability of this layer for describing service-oriented architectures of current or future manufacturing system, a more detailed description needs to be provided. Aiming to deal with these issues, this paper has two main contributions. On the one hand, a more detailed architecture definition of the Communication Layer of RAMI 4.0 is provided by applying the concepts of the ISO 42010. This means, stakeholders and their concerns are derived, views are created and domain-specific model kinds are provided. On the other hand, the created artifacts are evaluated towards their applicability as well as usability by making use of the Software Architecture Analysis Method (SAAM). By specifying a case study and typical application scenarios, the developed approach is validated towards its feasibility of describing complex industrial SOAs in the context of RAMI 4.0. The main goal of both contributions is thereby the provision of a more detailed reference architecture definition, which allows any stakeholder interested in industrial communication aspects the engineering of own systems based on RAMI 4.0. Thus, this goal is addressed in this paper by outlining the development of the Communication Layer architecture and a small example showing how to actually apply this architecture in industrial projects.

To address these aspects, the remainder of this paper is structured as follows: In Section II, the related work concerning RAMI 4.0, the ISO 42010 and industrial SOAs is explained in more detail. The pursued approach as well as utilized evaluation strategy based on the case study are thereby outlined in Section III. Subsequently, the next section delineates the development of the Communication Layer architecture itself, while its application is described and validated in Section V. Finally, in Section VI the results of the conducted study are summarized and a conclusion is given.

II. RELATED WORK

A. *Reference Architecture Model Industrie 4.0*

The already standardized reference model RAMI 4.0 has been developed by a conglomerate of three German associations, ZVEI, Bitkom and VDMA. In order to enable the discussion of Industry 4.0 systems, a specific coordinate system has been introduced within the three-dimensional cube. Each axis thereby addresses a particular aspect of the industrial system, while the layers represent the domain-specific viewpoints. Due to these characteristics, RAMI 4.0 spans over the complete value chain and encloses multiple industry sectors. In addition, several use cases are provided and well-known standards are integrated, making it possible to adopt contemporary manufacturing systems towards the concepts Industry 4.0. A special feature of this reference model is that it is defined as a SOA, meaning that application components provide services to the other components through a communication protocol over a network. This allows the integration of independent vendors, products, and technologies. In contrast, RAMI 4.0 needs to consider the different methodologies or proprietary solution in order to secure mutual systems engineering.

In more detail, the architecture itself is structured in ”Life Cycle & Value Stream”, ”Hierarchy Levels” and ”Interoperability Layers”, which are delineated in Figure 1. The vertical axis thereby deals with the integration of the components with a factory according to the well-known “automation pyramid”. By the term Hierarchy Levels, a guideline how to classify system components according to their application area is introduced. In order to do so, the following planes have been specified: Connected World (operations including participants outside the company), Enterprise (processes, services and infrastructures on company level), Work Centers (separation of dependencies between enterprise processes), Station (differentiation and aggregation of work units), Control Device (management and monitoring the manufacturing process), Field Device (sensors and actors used for the manufacturing process), and Product (physical devices). Contradictory, the horizontal axis deals with the assets and their life cycle, by dividing into four different states. The aim of this axis thus is to collect the data referring to the asset, which is accumulated during its usage throughout the whole life cycle. RAMI 4.0 proposes the following states: Type/Development (the prototype of the asset), Type/Maintenance (validation and verification of the prototype), Instance/Production (produced products of type) and Instance/Maintenance (products in usage).

In the end, the top-down arrangement of the layers provides the domain-specific viewpoints. Each layer thereby considers aspects of the system according to its appellation, making it possible to structure the whole industrial system. The Business Layer thus addresses business-related aspects and requirements while the Function Layer deals with the functions of the system. Data handling and their interchange is handled in the Information and Communication Layer, while the Integration enables the digitalization of physical components, which are depicted in the Asset layer.

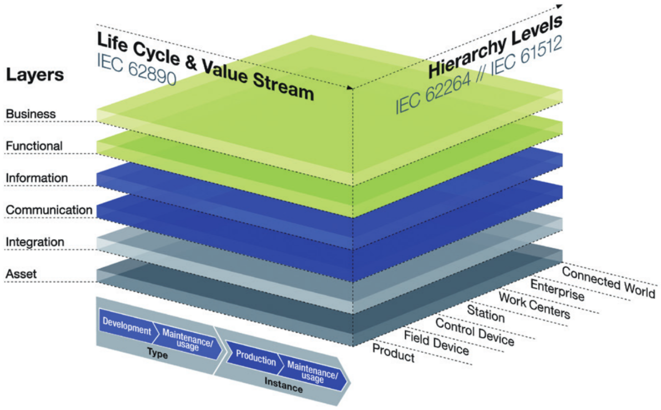

Fig. 1. Reference Architecture Model Industrie 4.0 (RAMI 4.0) [3]

B. *ISO 42010*

The ISO/IEC/IEEE 42010 International Standard [5] provides a concept on how to describe complex systems in terms of architecture. It can be used as basis for the development of architecture descriptions, architecture frameworks and architecture description languages. The main goal of this standard is to support the creation, analysis and maintainability of a system, which are growing in complexity. Additionally, the communication between the system's stakeholders is strongly encouraged by describing a standardized architecture. Out of these reasons, the standard defines all information an architecture framework shall include, explained in the following:

- Architecture framework identification
- Stakeholder identification
- Stakeholder concern identification
- Viewpoint and model kind provision
- Correspondence rules

In order to specify the architecture framework with all viewpoints, Architecture Description Languages (ADLs) are usually used within this standard. Those languages are needed to frame the concerns of the stakeholders in order to support architecture development through interdisciplinary systems engineering. According to the specifications, an example of such an ADL is the Unified Modeling Language (UML) or the Systems Modeling Language (SysML). Another important aspect of the ISO 42010 is the definition of a well-defined process aiming to support the architecture development task. Different process steps provide information on how to model the system when using the framework and the applied modeling language.

C. *Service-oriented Architectures*

Due to the integration of Internet of Things (IoT)-devices into systems, the importance of SOAs is constantly increasing. For example, the authors of [6] analyze such SOAs to describe the energy efficiency in Smart Buildings. As building management systems need to be able to monitor, control, analyze and manage the components used within a Smart Building, the integration of a large number of proprietary devices is becoming increasingly difficult. A SOA thereby helps to identify the interconnection and the data exchange between those devices by specifying the functionalities of each device as services.

More specifically targeting the Industry 4.0 domain, a detailed analysis of SOAs for manufacturing system is published in [7]. In this work, the Internet of Services is tried to be delimited and the effects on the manufacturing environment are investigated. The authors thereby conclude that the Internet of Services is one pillar of Industry 4.0, as each manufacturing element provides their functionality in form of a service. Thus, a SOA could deal with better characterizing this kind of manufacturing system.

A more detailed approach has been proposed in [8]. By defining Digital Twins of actual physical components, the minimal needed data to enable service-orientation in such a manufacturing system can be derived. The services themselves are analyzed towards their quality concerning time, money and product quality as well as the capabilities the service is able to fulfill. The needed data is thereby gathered from two use cases describing manufacturing plants by modeling the customers' orders and the products to be manufactured. While making the first steps of defining a SOA in this direction, the proposed approach appears to be promising when it comes to modeling the services of Digital Twins.

Other approaches dealing with service-oriented development of industrial systems also make clear that the development of SOAs in this area is needed. For example, another approach also uses RAMI 4.0 to retrofit original manufacturing systems towards Industry 4.0 [9], while the authors of [10] define a model-driven development process based on the Reference Model for Service Architecture (RM-SA). In addition, some publications also introduce new SOAs for Industry 4.0-based systems [11], [12]. While most of the mentioned approaches propose initial prototypes and frameworks in this area, they mostly solely address one single aspect of the manufacturing system or target a specific domain. In order to enable mutual engineering of current and future industrial systems including multiple stakeholders and considering a large variety of domains, a holistic and standardized approach needs to be available.

III. APPROACH

The goal of this paper is to develop a detailed architecture definition for the Communication Layer of RAMI 4.0. This will allow the creation of service-oriented system architectures of current or future industrial systems. As the Communication Layer is missing in specifications considering the official standard DIN SPEC 93145 [13], a more detailed definition needs to be provided. This is done with the help of the ISO 42010, an established architecture development methodology. As the creation of such an architecture in this area is a new topic to talk about, the development process is executed in small steps and iteration cycles. Thus, the Agile Design Science Research Methodology (ADSRM) appears to be a suitable approach when following such a strategy [14]. This method supports engineering tasks by specifying requirements, developing artifacts, applying examples and verify or validate the result. However, the entry point into the iteration cycle is typically a suitable case study. In this scenario, a case study of a manufacturer of copper-plated metal plates is applied. In order to integrate Industry 4.0-related aspects, the production process should be automated and the code to address the Programmable Logic Controller (PLC) should automatically be generated from the model. Particularly the Communication Layer needs to deal with different aspects of this system, as interconnections, technologies and provided services.

In order to validate the resulting SOA towards feasibility and applicability, the SAAM is applied [15]. This method aims to evaluate one or multiple architectures in terms of fulfilling the requirements and the actual purpose they are designed for. Special focus is thereby set on the users of the architecture, as they need to apply it for their purposes. Thus, in this paper, three different application scenarios have been defined, which are based on the previously mentioned case study. They are described as follows:

1) The network developer needs to gain information how to set up the Information and Communication Technology (ICT)-infrastructure from the architecture. Additionally, the interconnection between the components and the provided services are of importance.
2) The production planner wants to automatically perform tasks on the PLCs, according to previously calculated production parameters.
3) The component provider has to set up the components' interfaces and thus gathers information about provided services, technologies and exchanged data protocols.

IV. IMPLEMENTATION

In this section, the development and implementation of the Communication Layer architecture will be described in detail. This is done by first elaborating the architectural aspects of the ISO 42010 and adjusting them for the Communication Layer. Afterwards, a Domain Specific Language (DSL) is implemented utilizing the concepts of UML to allow the development of SOA-based architectures in the context of RAMI 4.0. At last, the development of the RAMI Toolbox, to apply the previously defined architecture, is illustrated.

A. Architecture definition

The first step of developing a detailed architecture for the RAMI 4.0 Communication Layer is the specification of views and model kinds in order to address the corresponding stakeholder concerns. Those artifacts are aligned to the ISO 42010 standard, which deals as a foundation for the architecture definition. To not exceed the scope of this paper, this application scenario introduces three specific stakeholders having interest in the Communication Layer architecture. The first stakeholder is described as network management of the company. The main goals are thereby ensuring the interconnection of the single departments or manufacturing units of the company by specifying the ICT network topology. This means, the architecture should provide all information about the needed network components and their physical or virtual connection. Subsequently, the second stakeholder is defined as the Supervisory Control and Data Acquisition (SCADA) supervisor. This stakeholder deals with managing the tasks for manufacturing machines based on the SCADA. As one of the main goals of Industry 4.0 is the automation of production processes, OPC UA provides the technical infrastructure to address the SCADA system with suitable protocols and information objects. Therefore, the SCADA supervisor needs to gain information about OPC UA objects within the architecture. Finally, the third stakeholder is the solution provider in general. As each of the system's components is fulfilling a functionality, this function has to be available as service to other components. Either they can directly execute the function or the are provided with the results when submitting a specific input. In order to address the three mentioned stakeholders as well as their concerns, the following views have been specified within the architecture of the Communication Layer:

- ICT Network Topology View
- Service View
- Communication Standard View

In order to ensure the modeling of systems according to the mentioned views, several model kinds have been defined. For applying the Interface View, the Object Management Group (OMG) open-source specification Service-oriented architecture Modeling Language (SoaML) has been integrated. Additionally, UML and SysML concepts like port or interface diagrams help to understand, how the services are provided from the respective components. As far as the ICT Network Topology View as well as the Communication Standard View is concerned, a particular DSL has been developed to enable modeling within those viewpoints.

B. DSL Adaption

After domain-specific stakeholder, concerns and viewpoints have been defined, the next step is to adapt the already existing DSL for developing industrial systems based on RAMI 4.0. This modeling language contains all elements for describing a system on each of the abstraction layers of the reference architecture. Thus, in this specific case, modeling elements for developing the Communication Layer architecture need to be defined. The metamodel is thereby derived by utilizing the concepts of UML. In consequence, two different adjustments have been made to the metamodel of RAMI 4.0. The first adaption is thereby considering the ICT Network Topology View by providing model kinds to contribute to the overall goal of this view. Thus, a deployment diagram has been extended with domain-specific aspects in order to describe an ICT network architecture. In this diagram, a number of network components are provided, like mobile devices, routers, switches, firewalls, servers and cloud-related assets. In order to interconnect those component, different network connection types as well as technologies are offered by the toolbox of this diagram. For example, wired or wireless connection types are provided.

The second adaption of the DSL is the creation of a specific OPC UA client element. This modeling element contains specifications for the identification of the PLC to be interconnected. Via a REST-interface, the communication between the client and the PLC is ensured. Additionally, a XML-based template file is added to the RAMI Toolbox. According to the attributes or values entered in the OPC UA element, the XML-file is adjusted. This allows to embed tasks for specific machines within this file, according to the previously modeled specifications. Afterwards, the configured file is transmitted via the REST-interface to the corresponding PLC, where the embedded task is executed. In conclusion, the mentioned process allows to address machines directly from the system architecture model, which significantly enhances the usability and automation potential within this area.

C. Tool Implementation

The last step to enable the development of SOA-based system architectures regarding the Communication Layer of RAMI 4.0 is the introduction of a specific tool. Better known by the term RAMI Toolbox, this software is especially designed to support the modeling of industrial system architectures in this area and thus provides all needed functionalities. By doing so, the RAMI Toolbox[1] itself is available as Add-In for the modeling environment of Enterprise Architect (EA). After initializing, the toolbox loads the metamodel and DSL elements within the UML profile to make them available for potential users. In order to increase the usability and automation potential, several functions are provided. In this specific scenario, the interface function between the model and the PLC is provided by the toolbox. This function is executed by highlighting an OPC UA client element and automatically generates the XML based on the attributes of this element. Subsequently, this file is directly transmitted to the PLC, which executes the task.

V. VALIDATION

This section deals with validating the developed Communication Layer architecture according to SAAM by applying the chosen application scenario. In this case study, as specified in Section III, the communication infrastructure of a manufacturer of copper-plated metal plates is analyzed in detail. By automatizing the process of code generation, different services need to be provided, the ICT infrastructure needs to be available and the resulting code needs to be sent to the PLC via OPC UA. As modeling all functions, requirements and technical aspects would exceed the scope of this paper, a number of specified functions is used. More precisely, functions like *calculate finale reinforcement speed*, *choose model* and *measure borehole size* are used in this scenario. In order to fulfill those functions, different components are introduced, which either provide the service or make use of it. The first scenario therefore deals with the Service View of the architecture and how those services are interchanged. This is done by modeling the components via SoaML with regard to domain-specific aspects. As Figure 2 depicts, the data exchange between the services is executed with the help of ports and interfaces. For example the measured date is provided by a measurement sensor and used by the velocity controller, where the data is transmitted via Near Field Communication (NFC). This component however calculates and provides the final reinforcement speed, which is provided to the velocity switch as well as the code controller. Finally, the model switch chooses the right model for the machine code and passes this code to the variable writer actor via Ethernet. In a lower abstraction level, the interfaces and ports are described in more detail, so that the solution provider is able to perform the right technical implementation.

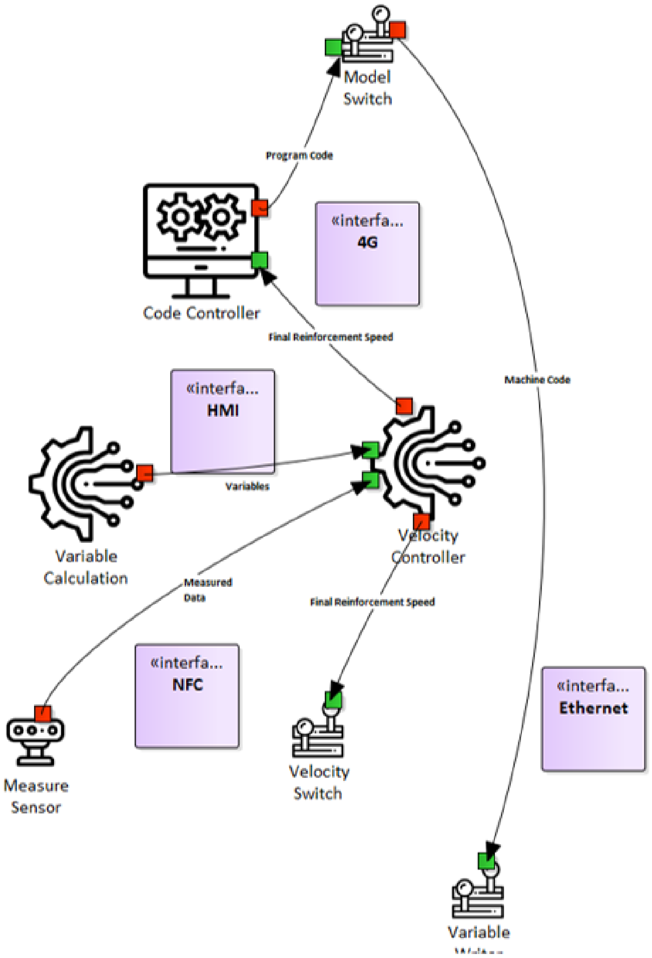

Fig. 2. Service View of the application scenario

As far as the ICT infrastructure is concerned, the DSL-based diagram is applied. The result is thereby depicted in Figure 3. This figure indicates that the production planning department is responsible for ordering the copper via the internet from any available supplier. After purchasing the copper, it is transferred to the storage area, which is connected via Ethernet. Additionally, the Enterprise Resource Planning (ERP) server as well as the copper-plating machine are interconnected within the same network. This allows all components to query the current copper stock and thus plan the production. In this diagram, all network components like firewalls, routers and switches are modeled. The blue lines furthermore represent Ethernet connections, while the red lines define wireless connections.

At last, the OPC UA clients and servers are modeled within the architecture of the Communication Layer. As already explained in Section IV, this will enable the control of PLCs directly from the model. In this example, the PLC controlling the velocity of the copper-plating machine is addressed with an XML-file. This file is generated with the help of the RAMI Toolbox and submitted via a REST-interface.

---

[1] The RAMI Toolbox is publicly available for download at http://www.rami-toolbox.org/download

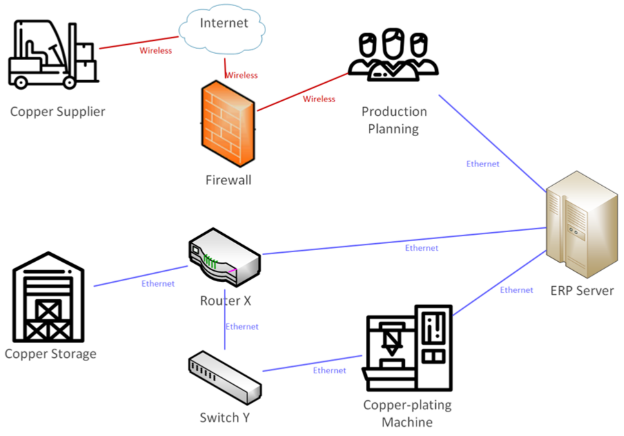

Fig. 3. ICT Network View of the application scenario

In this case, the production planning department is able to control the production process directly from the model, with the Communication Layer architecture model as central communication unit.

The scenario-based evaluation of the resulting SOA indicates, that all need information is provided by the architecture of the Communication Layer. Each of the three stakeholders is able to gain the aspects of its interest, when considering the copper-plating case study. However, with the superficial use case like the one applied in this paper, the most important attribute to investigate is the feasibility of the architecture. Thus, with regard to the results of this paper, a first version of a SOA enabling systems engineering on the Communication Layer of RAMI 4.0 is introduced. The RAMI Toolbox thereby provides all needed artifacts in order to address the applicability and usability of the architecture. However, while those attributes are also important aspects to consider, they should be further evaluated with a more sophisticated case study in the next iteration of ADSRM.

VI. CONCLUSION & FUTURE WORK

As manufacturing processes are drifting away from product-orientation towards the provision of services, the need to provide a SOA becomes obvious. Thus, in this paper, the development of an SOA in order to more precisely define the Communication Layer of RAMI 4.0 is outlined. As the standardized definition is lacking in specifications and thus hinders the applicability of RAMI 4.0, the results of this paper provide a more detailed architecture definition, which supports systems engineers in their task of modeling complex industrial systems. Therefore, in Section IV, the development of the architecture itself is illustrated, which makes use of ISO 42010 concepts like the elaboration of viewpoints. Additionally, a specific DSL is implemented to address domain-specific aspects. The resulting architecture is then applied in Section V. By utilizing the software architecture analysis methods of SAAM and a real-world case study, the feasibility, usability and applicability of the architecture is investigated. The resulting artifact is an applicable architecture of the RAMI 4.0 Communication Layer, allowing the engineering of complex industrial systems in this area for the first time.

However, the proposed architecture should not be considered a fully specified ready-to-use methodology but rather a step into the right direction. To further enhance the outcome of this paper, several future projects could be implemented. For example, the architecture should be evaluated with a more sophisticated use case to ensure the applicability in a larger application area and for more complex systems. Additionally, the interfaces to the other layers of RAMI 4.0 need to be defined, which enables holistic and mutual MBSE of large-scale manufacturing systems.

ACKNOWLEDGMENT

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development and the Christian Doppler Research Association as well as the Federal State of Salzburg is also gratefully acknowledged.

REFERENCES

[1] R. M. Claude Jr and D. A. Horne, "Restructuring towards a service orientation: the strategic challenges," International Journal of Service Industry Management, vol. 3, no. 1, 1992, pp. 0–0.
[2] D. Bauer, D. Stock, and T. Bauernhansl, "Movement towards service-orientation and app-orientation in manufacturing it," Procedia CIRP, vol. 62, 2017, pp. 199–204.
[3] Bitkom, VDMA, ZVEI, "Umsetzungsstrategie Industrie 4.0, Ergebnisbericht der Plattform Industrie 4.0," ZVEI, 2015.
[4] C. Binder, C. Neureiter, and A. Lüder, "Towards a domain-specific approach enabling tool-supported model-based systems engineering of complex industrial internet-of-things applications," Systems, vol. 9, no. 2, 2021.
[5] International Organization for Standardization, "ISO/IEC/IEEE 42010: Systems and software engineering – architecture description," 2011.
[6] L. Mendoza-Pitti, H. Calderón-Gómez, M. Vargas-Lombardo, J. M. Gómez-Pulido, and J. L. Castillo-Sequera, "Towards a service-oriented architecture for the energy efficiency of buildings: A systematic review," IEEE Access, vol. 9, 2021, pp. 26 119–26 137.
[7] J. Z. Reis and R. F. Gonçalves, "The role of internet of services (ios) on industry 4.0 through the service oriented architecture (soa)," in IFIP International Conference on Advances in Production Management Systems. Springer, 2018, pp. 20–26.
[8] F. Schnicke, T. Kuhn, and P. O. Antonino, "Enabling industry 4.0 service-oriented architecture through digital twins," in European Conference on Software Architecture. Springer, 2020, pp. 490–503.
[9] J. D. Contreras, J. I. Garcia, and J. D. Pastrana, "Developing of industry 4.0 applications." International Journal of Online Engineering, vol. 13, no. 10, 2017.
[10] B. Liu, T. Glock, V. P. Betancourt, M. Kern, E. Sax, and J. Becker, "Model driven development process for a service-oriented industry 4.0 system," in 2020 9th International Conference on Industrial Technology and Management (ICITM). IEEE, 2020, pp. 78–83.
[11] F. Z. El Kho and N. Aknin, "Thing-based service-oriented architecture for industry 4.0," in International Conference on Artificial Intelligence and Symbolic Computation. Springer, 2019, pp. 289–300.
[12] J. Al-Jaroodi, N. Mohamed, and I. Jawhar, "A service-oriented middleware framework for manufacturing industry 4.0," ACM SIGBED Review, vol. 15, no. 5, 2018, pp. 29–36.
[13] DIN SPEC, "91345," RAMI 4.0, 2016.
[14] K. Conboy, R. Gleasure, and E. Cullina, "Agile design science research," in International Conference on Design Science Research in Information Systems. Springer, 2015, pp. 168–180.
[15] R. Kazman, G. Abowd, L. Bass, and P. Clements, "Scenario-based analysis of software architecture," IEEE software, vol. 13, no. 6, 1996, pp. 47–55.