Using a model-based engineering approach for developing Industrial Internet of Things applications

Christoph Binder
Center for Secure Energy Informatics
Salzburg University of Applied Sciences
Urstein Sued 1, A–5412 Puch/Salzburg, Austria
christoph.binder@fh-salzburg.ac.at

Dieter Draxler
Center for Secure Energy Informatics
Salzburg University of Applied Sciences
Urstein Sued 1, A–5412 Puch/Salzburg, Austria
dieter.draxler@fh-salzburg.ac.at

Christian Neureiter
Center for Secure Energy Informatics
Salzburg University of Applied Sciences
Urstein Sued 1, A–5412 Puch/Salzburg, Austria
christian.neureiter@fh-salzburg.ac.at

Goran Lastro
Center for Secure Energy Informatics
Salzburg University of Applied Sciences
Urstein Sued 1, A–5412 Puch/Salzburg, Austria
goran.lastro@fh-salzburg.ac.at

Abstract—**Industrial Internet of Things (IIoT), a new way of developing manufacturing systems and their participants, better known as Cyber-physical Systems (CPS), is a term that is promoted by the emergence of the so-called fourth industrial revolution. This new trend in industrial manufacturing offers new automation possibilities with the goal to optimize operational efficiency, minimize costs and realize new business models. However, accompanied with those new opportunities, on the other hand engineering such systems and their architectures has become a complex and difficult task. Having recognized this issue, the Reference Architecture Model Industrie 4.0 (RAMI 4.0) has been introduced in order to provide viewpoints for structuring an Industry 4.0 based system according to the different concerns to address. Although providing a standardized framework, at the current point of view it is difficult to describe a detailed system architecture, since this reference model is missing formalizations and common methods. Therefore, this paper proposes a possibility of aligning the model-based engineering methods introduced by the Software Platform Embedded Systems (SPES) with the architectural concepts of RAMI 4.0 in order to close the aforementioned gap and provide well-defined methodology in order to develop current and future industrial systems. To achieve this, similarities between both approaches are analyzed and compared with each other and a architecture definition based on the ISO 42010 is given, which is subsequently evaluated by the application of a real-world case study.**

Index Terms—**System Architecture, Industrial Internet of Things (IIoT), Reference Architecture Model Industrie 4.0 (RAMI 4.0), Domain-specific Systems Engineering (DSSE), Software Platform Embedded Systems (SPES)**

I. INTRODUCTION

Contemporary manufacturing companies are constantly forced to further develop their ways of value creation in order to remain competitive. With new technological advances from research and development, this process is constantly supported by providing new methods or technologies. One of the main improvement resulting from this is the emergence of the Industrial Internet of Things (IIoT). This trend describes the alignment of the Internet of Things (IoT) to the industrial area aiming to pursue new business models as well as optimize production by reducing expenses at the same time, widely known by the term "Industry 4.0" [1]. Furthermore, the essential technology driver ensuring the application of the IIoT are so called Cyber-physical Systems (CPS). As explained in [2], those units are mainly intelligent components of a manufacturing process or parts of a value creation system, which are able to find the economically most valuable decision on their own. This means, the interconnection of those CPS within an industrial system forms a service-oriented value creation network facing challenges like self-sustaining production or real-time information handling [3]. As a result, these changes result in a new kind of automation driven industry, which drifts away from the original product-orientation towards technology-oriented services [4].

Having recognized this transformation, several leading German associations proposed the Reference Architecture Model Industrie 4.0 (RAMI 4.0), a three-dimensional cube for structuring a complex industrial system based on a Service-oriented Architecture (SOA) [5]. Specialized for the application in the manufacturing area, the reference architecture and its methods are specified in the standard DIN SPEC 91345 [6]. However, since the just mentioned specifications and the underlying norm are only a theoretical concept, suitable applications using its methods need to be provided in order to consolidate its utilization. Thus, the authors of this paper previously introduced a Domain Specific Language (DSL) [7] as well as an associated development process [8], summarized and provided by a software tool called RAMI Toolbox.

By now, the RAMI Toolbox has been utilized in different projects with the intention to develop architectures of different kinds of systems. For example, one of the application scenarios is introduced in [9], where the functional architecture of a car engine is developed in detail based on the specifications given by RAMI 4.0. However, one of the main insights of the mentioned work is the missing granularity level in the RAMI 4.0 cube itself. Considering an industrial system consisting of several CPS as a complex System of Systems (SoS), engineering those needs to consider established design principles used by Model Based Systems Engineering (MBSE), like divide and conquer or separation of concerns [10]. Explained in more detail, although RAMI 4.0 is trying to address these principles, the cubic layout solely allows the description of production lines on the highest level, providing only a black-box perspective on its components. This hinders the accurate model-based development and implementation of actual industrial manufacturing systems.

Taking this into further consideration, when it comes to treat a component on different granularity layers, RAMI 4.0 needs to be extended by additional methodologies. For example, Software Platform Embedded Systems (SPES) seems to be one of the most promising methods enabling MBSE in several domains [11]. Thus, the methods provided by this approach seem to be a suitable concept for dealing with this issue. Therefore, this paper introduces two major contributions. First the RAMI 4.0 layers are further refined by mapping them to the matrix layout of SPES. Consequently, a detailed architecture definition is elaborated by making use of the ISO 42010 [12]. In the end, the developed artefacts are evaluated by applying a real-world case study. The result of this approach would not only enhance the applicability of RAMI 4.0, but could also takes industrial systems engineering one step further by combining two broadly accepted methodologies and turn them into a common foundation for developing future manufacturing systems.

To address these aspects, this contribution is structured as following: In Section II an overview of RAMI 4.0, SPES and several methods enabling industrial MBSE is given. Next, the approach to challenge the mentioned problem is stated in Section III. The mapping of both approaches and the architectural description is mentioned in Section IV, whose applicability is demonstrated with an actual industrial use case 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

The goal of RAMI 4.0 is to enable the discussion of an Industry 4.0 system based on domain-specific viewpoints. The three-dimensional model has been mainly developed to create a common understanding and a mutual basis. Due to the big influence of its creators on the German industry, the reference architecture encloses multiple industry sectors and spans over the complete value chain. Moreover, a developed system can be seen as whole because of the integration of well-known standards and use cases related to Industry 4.0. On the one hand this includes all interconnections and sequences of events, on the other hand the possibility of a detailed consideration of its single parts is given. Doing so, the architecture itself is structured in ”Life Cycle & Value Stream”, ”Hierarchy Levels” and ”Interoperability Layers”, which are delineated in Figure 1.

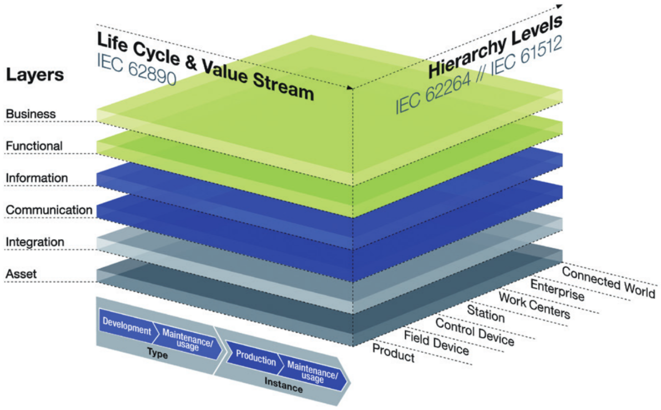

Fig. 1. Reference Architecture Model Industrie 4.0 (RAMI 4.0) [13]

This means, that the horizontal alignment of RAMI 4.0 gives more insights into the life-cycle and the different states of an asset. By falling back to the criteria introduced in the standard IEC 62890 [14], the purpose of this axis is to collect all data, which is accumulated during its usage throughout the whole life-cycle. By distinguishing between type and instance as well as development and usage, a system component can be described from its idea to its disposal.

Furthermore, the vertical integration within a factory is realized by the second axis, the so-called Hierarchy Levels. Based on the standards IEC 62264 [15] and IEC 61512 [16], better known under the term *automation pyramid*, a guideline for classifying the component according to its application area is provided. 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). The last axis of the three-dimensional cube introduces six different layers, which deal as viewpoints for structuring a system and providing different perspectives on it. In more detail, the viewpoint located at the highest, called Business Layer, defines the context and boundaries of a system by providing tools for modeling the business perspective. Based on this, the future development of the system is initiated by generating functions for fulfilling the requirements in the Function Layer. Next, within the two viewpoints located in the middle, the Information Layer explains which information is exchanged while the Communication Layer defines how. Moreover, the Integration Layer enables the Digital Twin representation and the Asset Layer illustrates the physical representation [17].

B. Software Platform Embedded Systems (SPES)

As the name assumes, the SPES modeling framework is used for model-based development of different kinds of systems. The SPES modeling framework is a set of tools for enabling model-based development of different kinds of systems in several areas. By doing so, the framework addresses challenges originating from different application domains like automation, healthcare and automotive amongst others. Based on certain requirements and principles, which are defined in [11], a new way of thinking in regard to systems engineering is proposed. For example, the concept assumes that the characteristics of the system to develop should be derived from the requirements withing the specific application domain.

To fulfill the aforementioned objectives, the SPES framework inherits two major concepts, the so-called *Abstraction-Layers* as well as *Views and Viewpoints*. The combination of those form a two-dimensional engineering space, as proposed in [18]. More precisely, the *Viewpoints* are separated into four different sections in the horizontal axis, which are *Requirements Viewpoint, Functional Viewpoint, Logical Viewpoint* and *Technical Viewpoint*, whereas each viewpoint is realized by different templates or methods. However, the vertical alignment introduces the different abstraction layers according to the *divide and conquer* principle.

C. Domain Specific Systems Engineering

Since the application domains summarized under the umbrella term "Industry 4.0" are far-reaching and difficult to handle, systems in this area are mostly complex and need to be engineered with the right choice of methods. As a contemporary industrial system usually being a SoS, the concept of MBSE has proven to be a suitable way to consider every aspect included while creating the architecture of such a system [19]. This means, a generic approach most likely hinders the modeling of such a system with dynamic structures and changing conditions. Thus, an approach providing a DSL and a suitable modeling process based on the Model Driven Architecture (MDA) has been proposed earlier. In addition, other contributions making use of similar concepts have been published in recent years. For example, several authors introduced potential ways of modeling a system or single aspects of it utilizing RAMI 4.0 as reference architecture and applying different modeling languages like Unified Modeling Language (UML), Systems Modeling Language (SysML) or even defining a DSL [20]–[22]. In contrast, the authors of [23] describe an approach for the development of IIoT applications by making use of the concepts of Industrial Internet Reference Architecture (IIRA). A special feature of their work is the mapping of the IIRA viewpoints to those of the Unified Architecture Framework (UAF), which enables MBSE by applying the extensive possibilities for model-based systems development of this framework.

III. APPROACH

As already mentioned, the goal of this contribution is to combine the concepts of RAMI 4.0 with those coming from SPES in order to provide a more heterogenic and detailed architecture of current and future industrial systems. Since both approaches provide a layered architecture, similarities can be identified and a common link needs to be established. On the other hand, it is also significant to find a solution for closing the recognized gaps. This means, in order to enable the development of manufacturing on different abstraction levels, the goal is to combine both frameworks for profiting from their respective advantages. This is done by developing some kind of umbrella architecture, where the viewpoints of both frameworks are interconnected and aligned to the ISO 42010 as well as made applicable by utilizing the top-down development process of the modeling paradigm MDA [24].

Since this is the first time these two frameworks are compared with each other and the accompanied unpredictability of the outcome, an agile methodology for approaching this issue needs to be applied. This is furthermore underlined by the high rate of change in terms of used methods in the area of IIoT. Hence, the proposals of the Agile Design Science Research Methodology (ADSRM) are tailored to such dynamic application scenarios. This agile methodology for application-related research and development introduces five process steps, which also deal as so-called entry points for allowing to enter the development cycle. This means, flexible development is promoted by defining small iteration cycles and the possibility to perform changes within each process step. The whole process of ADSRM is supported by so-called exploratory case studies. Assuming such a case study to be the entry point for development of the functional architecture, it deals as reference point for the other phases. In more detail, the requirements and the artefacts to develop are derived from the findings of the case study definition. Afterwards, the case study itself is practically implemented and evaluated by applying the previously developed items [25].

In this example, the designated case study makes use of an industrial robot for building cars with modular manufacturing units. By doing so, all data and information is coming from a company partner and can therefore be considered as a real-world industrial use case. However, more details and a specific step by step illustration of the elaborated case study is given more precisely in Section V.

IV. IMPLEMENTATION

A. Mapping RAMI 4.0 to SPES

The first step of interconnecting RAMI 4.0 with SPES has been set in [26], where architecture frameworks of different domains have been mapped to the SPES framework in order to enable cross-domain modeling. However, as this approach is targeted to the industrial area, special focus on RAMI 4.0 is given in this paragraph. Following this principle, as assumed by their names, the mapping of the Business Layer to the Requirements Viewpoint is more or less straightforward, as they can be transformed one-on-one. The same method can be applied to the Function Layer by mapping it to the equally called Function Viewpoint of SPES. However, considering the Information and Communication Layer of RAMI 4.0, the SPES matrix does not provide a viewpoint for those. Therefore, the mapping process for interconnecting the two modeling frameworks has to be enhanced. In this case, the two RAMI 4.0 layers are combined and then mapped to the

Technical Viewpoint of SPES. This leads back to the fact that the interfaces of the single components or the protocols for exchanging data are of technical nature. This means, that the mapping is uni-directional, since raveling out the information of the Technical Viewpoint would be considered a increasingly challenging task. Furthermore, as the Asset Layer also would find its place in this SPES viewpoint, additional information is appended to the technical representation, which hinders the mentioned process even more. At last, as the Integration Layer inherits the Digital Twin representation with all Asset Administration Shell (AAS) information, it contains the data of several viewpoints. Thus, this RAMI 4.0 layer reaches from the Functional Viewpoint across the Logical Viewpoint up to the Technical Viewpoint of SPES.

B. Architecture Definition

The first part of applying the ISO 42010 for describing a system architecture is to provide one or more viewpoints for each of the stakeholder concerns. Therefore, firstly the stakeholders and their concerns with regard to functional architectures are elaborated in the first place. To give a short overview of the results of this process, some examples are given in the following. The requirements engineer has interest in accurately formalizing them, while the function developer or the process engineer are concerned in the detailed functional description including in- and outputs. Moreover, the manager's concern is to fulfil the customers' requirements, whereas the network administrator needs a detailed specification of all technical components. As the viewpoints have been already defined in the definition of the SPES framework, the next step is to define model kinds for each viewpoint in order to realize the architectural view. Summarized, an overview of the specified model kinds for each viewpoint is given.

- Requirements Viewpoint:
    1) Context Model
    2) Goal Model
    3) Scenario Model
    4) Requirements Model
- Function Viewpoint:
    1) FAS Model
    2) Black Box Model
    3) White Box Model
- Logical Viewpoint:
    1) Concept Model
- Technical Viewpoint:
    1) Block Definition Model
    2) Internal Block Model

Explained in more detail, the Requirements Viewpoint makes use of four different model kinds. The Context Model is used to surrounding systems as well as their in- and output, while the Goal Model deals with identifying the stakeholders and their interests in the system. The outcome of both models is summarized in order to identify specific scenarios in the Scenario Model. However, as the requirements are the most important part for fulfilling the task of the equally named viewpoint, an own model has been defined for specifying them. Subsequently, in the Function Viewpoint, the first step is to elaborate the system functions based on the previously specified requirements. Therefore the Functional Architecture for Systems (FAS) methodology is taken for use. The resulting functions are the described in more detail in the Black Box as well as the White Box View. The third column of SPES, the Logical Viewpoint inherits a model where the first realization of the system is delineated. Thus, the Concept Model contains specific elements for describing logical components that fulfill the functions on the one hand as well as digitalizing the mechanical system components according to the IIoT concepts on the other hand. At last, in the Technical Viewpoint, the actual real-world system with all its components is modeled with the help of the block definition diagram or the internal block diagram provided SysML and their corresponding models.

C. Integration into the RAMI Toolbox

The last step to complete the combination between both approaches is to integrate the particularities of SPES into the RAMI Toolbox. Thus, first the typical matrix layout of SPES needs to be available for users in order to structure the architecture of a system according to it. To do so, an additional step-through user interface has been developed, as seen in 2. Thereby, the respective rows and columns represent the matrix of SPES with its viewpoints and various abstraction levels, while the colors indicate the RAMI Layers. According to the specifications in the previous section, a different modeling task has to be fulfilled in each of the squares. For example, in the Requirements Viewpoint only a certain number of models are available to address the stakeholder concerns. To ensure this, an *Add Model* function appears when one of the squares is clicked. Additionally, an *Allocate* and a *Decompose* function is added in the same step. The purpose behind this is enabling semi-automatic model transformations, as introduced by MDA. In this case, a further window appears where the trace between the elements can be added to the model.

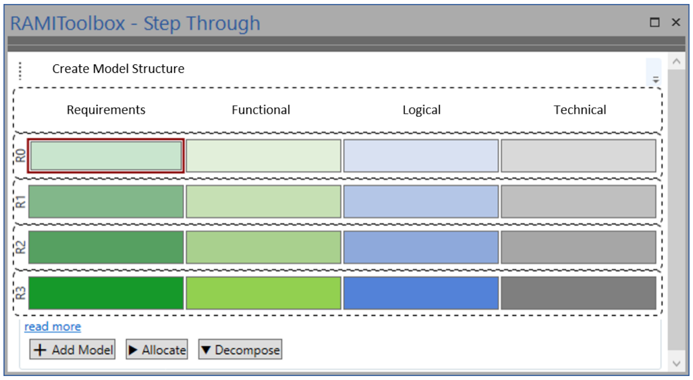

Fig. 2. SPES Window realized in the RAMI Toolbox

V. APPLICATION

In the following, the modeled case study will be explained in detail. As explained earlier, an industrial robot is utilized in this scenario. Since this example is provided by a company partner, business models, the system context as well as the requirements are derived from an actual manufacturing system. Taking this into further consideration and implied by its name, the goal of the Requirements Viewpoint is to elaborate the requirements of the system to develop. Since the Business Layer of RAMI 4.0 is mapped to this viewpoint and the Computation Independent Model (CIM) of MDA are applied, the system is described on a higher level perspective to be understood by non-technical stakeholders. Thus, first the system context is modeled in order to understand input and output of it. In this case, the robot to develop can be seen as superordinate function with an input state and translational movement as output. Furthermore, interferences like a wall or disturbances like counterforce are also considered. The next step is to specify the business actors and their concerns into the system. This is realized by the goal model and the business case of moving the robot with a high reliability. Subsequently, different scenarios of the business case are modeled, which are realized by high level use cases. Those scenarios represent possible solutions and set the direction for further modeling activities like elaborating the actual requirements the robot needs to fulfil. In this example, the requirements are collected by an engineer and include aspects like processing quantified parameters or specifying algorithms for the Artificial Intelligence (AI). A detailed overview of the requirements on granularity level 0 is represented in Figure 3.

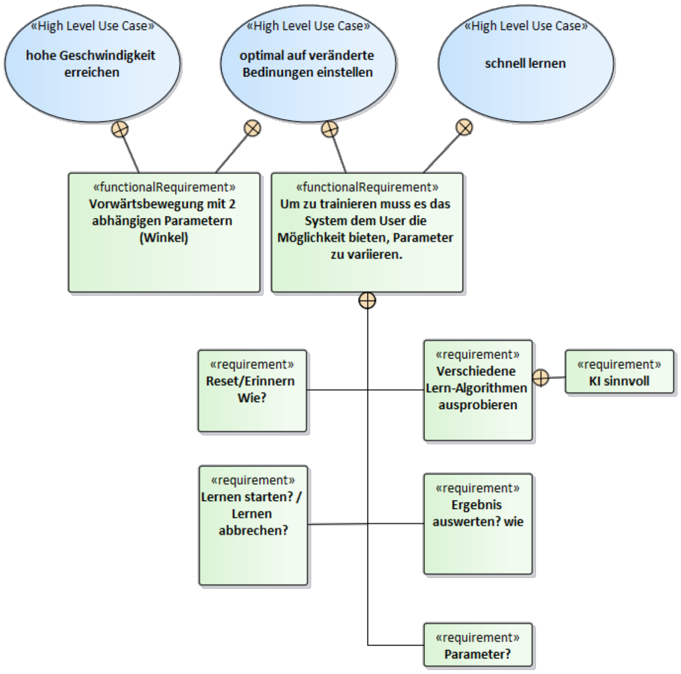

Fig. 3. Requirements Model

After defining the requirements, the next step is to specify actual system parts in order to fulfil the requirements. This is done by developing functions that carry out the required specifications. However, as one function is distributed over many physical elements and one element offers more than one function, the proposals of the FAS methodology are used to link those system parts together. As this is precisely described in [9], this will not be explained any further in this contribution. Consequently, in the Logical Viewpoint, the concept model is applied to find technical solutions for realizing the actual system components. In this case study and according to the requirements as well as the system functions, a robot with a double pendulum or a crawler can be applied on granularity level 0. In the last step, the actual technical component is modeled according to the specification of the technical architecture or the Platform Specific Model (PSM) of MDA, as seen in Figure 4. In this case, the modeling language SysML is used in order to describe the interfaces, data model standards or protocols of the system component as well as the details of the element itself. The just mentioned development process is tailored to the highest abstraction level of a system. However, since the matrix layout of SPES provides the same Viewpoints for all granularity levels, the approach is quite the same es the on explained in detail. The main difference is that the system of interest varies between the different levels. This means, each technical component defined in the upper layer becomes a new system to develop in the bottom layer, beginning with the Requirements Viewpoint.

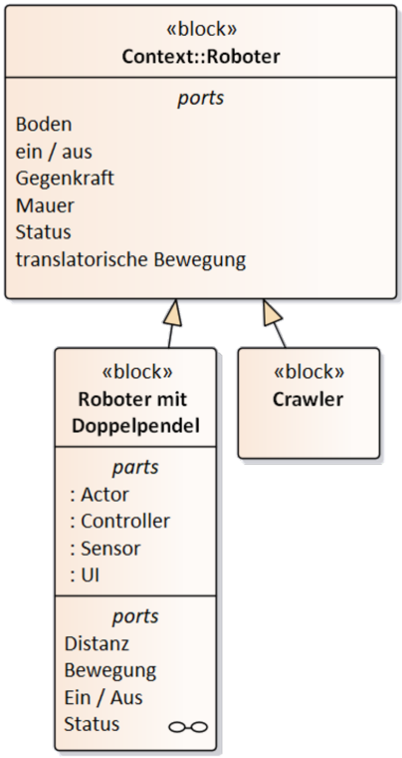

Fig. 4. Technical Architecture

Following this principle, a SoS can be developed on any number of depths needed by elaborating requirements, functions and technical aspects of each component as well as its subcomponents. As explaining the development of a subcomponent on a lower level would mostly be redundant and therefore exceeds the scope of this paper, more details can be found independently in the model of the case study[1].

VI. CONCLUSION & FUTURE WORK

The approach proposed in this paper deals with combining two well-known and established frameworks for developing architectures of industrial systems, RAMI 4.0 and SPES. While both reference architectures look good on paper and are already established in the community, it is still difficult to apply them for real industrial applications. A reason for this could be the missing specifications in each of the frameworks and a non-well-defined development process. In order to close this gap, the similarities between the mentioned approaches are analyzed and a method making use of the respective advantages is introduced in this contribution. By doing so, first both architectural frameworks are mapped to each other with the help of the modeling paradigm MDA and a specific architecture definition is elaborated. Thus, this contribution has to be seen as a first step into the right direction providing only a superficial perspective for evaluating the approach and its implementation with the RAMI Toolbox by using a real-world case study. However, a more extensive analysis compared to other methods and the provision of detailed results has to be done with a more sophisticated case study.

The outcome of this work can contribute to a lot of other research projects. For example, after enhancing the applicability of RAMI 4.0, a new possibility of modeling more sophisticated use cases or application scenarios is given. As at the current point of view and with the current technology state only architectures of superficial industrial case studies could have been developed, this will open the door for further refinements. One of those refinements could be the integration of SysML throughout all areas of the RAMI 4.0 metamodel. Another aspect identified by working on this project is the difference between distinguishing between product and production. In particular, the SPES concepts work well if a smart product like a CPS is developed. However, if the whole production chain needs to be described, this approach reaches its limits. As per definition of RAMI 4.0 and IIoT both of the mentioned scenarios need to be considered, further research has to be done in order to advance the result of this contribution, which is done in the next iteration steps of ADSRM.

ACKNOWLEDGMENT

The support for valuable contributions of LieberLieber Software GmbH and successfactory consulting group is gratefully acknowledged. The financial support by the Federal State of Salzburg is also gratefully acknowledged.

---

[1] A click-through model is available at http://www.rami-toolbox.org/UseCaseRobot

REFERENCES

[1] A. Gilchrist, *Industry 4.0: the industrial internet of things*. Springer, 2016.

[2] G. Schuh, T. Potente, C. Wesch-Potente, and A. Hauptvogel, "Sustainable increase of overhead productivity due to cyber-physical-systems," 2013.

[3] E. Mueller, X.-L. Chen, and R. Riedel, "Challenges and requirements for the application of industry 4.0: a special insight with the usage of cyber-physical system," *Chinese Journal of Mechanical Engineering*, vol. 30, no. 5, p. 1050, 2017.

[4] 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.

[5] ZVEI - Zentralverband Elektrotechnik- und Elektronikindustrie e.V., "Industrie 4.0: Das Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0)," ZVEI, Tech. Rep., 2015.

[6] Deutsches Institut für Normung, "DIN SPEC 91345: RAMI 4.0," 2016.

[7] C. Binder, C. Neureiter, G. Lastro, M. Uslar, and P. Lieber, "Towards a standards-based domain specific language for industry 4.0 architectures," in *Complex Systems Design & Management*, E. Bonjour, D. Krob, L. Palladino, and F. Stephan, Eds. Springer International Publishing, 2019, pp. 44–55.

[8] C. Binder, C. Neureiter, and G. Lastro, "Towards a Model-Driven Architecture Process for Developing Industry 4.0 Applications," *International Journal of Modeling and Optimization*, vol. 9, no. 1, pp. 1–6, 2019.

[9] C. Binder, D. Draxler, C. Neureiter, and G. Lastro, "Towards a model-centric approach for developing functional architectures in industry 4.0 systems," 2019.

[10] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," in *2016 49th Hawaii international conference on system sciences (HICSS)*. IEEE, 2016, pp. 3928–3937.

[11] K. Pohl, M. Broy, H. Daembkes, and H. Hönninger, *Advanced Model-Based Engineering of Embedded Systems*. Springer, 2016.

[12] International Organization for Standardization, "ISO/IEC/IEEE 42010: Systems and software engineering – architecture description," 2011.

[13] Bitkom, VDMA, ZVEI, "Umsetzungsstrategie Industrie 4.0, Ergebnisbericht der Plattform Industrie 4.0," *ZVEI*, 2015.

[14] International Electrotechnical Commission, "IEC 62890: Life-cycle management for systems and products used in industrial-process measurement, control and automation," 2016.

[15] ——, "IEC 62264: Enterprise-control system integration," 2016.

[16] ——, "IEC 61512: Batch control," 2001.

[17] M. Hankel and B. Rexroth, "The Reference Architectural Model Industrie 4.0 (RAMI 4.0)," *ZVEI*, 2015.

[18] K. Pohl, H. Hönninger, R. Achatz, and M. Broy, *Model-based engineering of embedded systems: The SPES 2020 methodology*. Springer Science & Business Media, 2012.

[19] P. Micouin, *Model Based Systems Engineering: Fundamentals and Methods*. John Wiley & Sons, 2014.

[20] M. Arantes, R. Bonnard, A. P. Mattei, and P. de Saqui-Sannes, "General architecture for data analysis in industry 4.0 using sysml and model based system engineering," in *2018 Annual IEEE International Systems Conference (SysCon)*. IEEE, 2018, pp. 1–6.

[21] R. Sharpe, K. van Lopik, A. Neal, P. Goodall, P. P. Conway, and A. A. West, "An industrial evaluation of an industry 4.0 reference architecture demonstrating the need for the inclusion of security and human components," *Computers in Industry*, vol. 108, pp. 37–44, 2019.

[22] T. H.-J. Uhlemann, C. Lehmann, and R. Steinhilper, "The digital twin: Realizing the cyber-physical production system for industry 4.0," *Procedia Cirp*, vol. 61, pp. 335–340, 2017.

[23] A. Morkevicius, L. Bisikirskiene, and G. Bleakley, "Using a systems of systems modeling approach for developing industrial internet of things applications," in *2017 12th System of Systems Engineering Conference (SoSE)*. IEEE, 2017, pp. 1–6.

[24] W. A. Brown, "Model Driven Architecture: Principles and Practice," *Software and Systems Modeling*, vol. 3, no. 4, pp. 314–327, 2004.

[25] 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.

[26] B. Brankovic, C. Binder, D. Draxler, C. Neureiter, and G. Lastro, "Towards a cross-domain modeling approach in system-of-systems architectures," in *Complex Systems Design & Management*. Cham: Springer International Publishing, 2020, pp. 164–175.