Towards a Model-Driven Architecture Process for Developing Industry 4.0 Applications

Christoph Binder, Christian Neureiter, and Goran Lastro

Abstract—**Emerging technologies in the industrial area lead to continuously increased complexity concerning systems development. Varying approaches dealing with the same problem generate a number of heterogeneous solutions instead of concentrating on a mutual toolset in order to provide a common basis. Having recognized this problem, the German industry introduced the Reference Architecture Model for Industry 4.0 (RAMI 4.0), proposed in the standardized technical specification DIN SPEC 91345. Providing a three-dimensional model on how to structure industrial systems, the starting point for the discussion on how to deal with the upcoming complexity has been set. However, due to the current state of research, only the frame to work in has been specified. In this paper the idea of following an approach based on Model-Driven-Architecture (MDA) in order to develop future industrial systems is introduced. To achieve this, firstly the concepts of RAMI 4.0 are analyzed and a detailed description regarding their applicability for developing specific architectures is given. The approach itself and its application are demonstrated by a real-world case study, which is created with the help of the RAMI Toolbox.**

Index Terms—**Model driven architecture (MDA), RAMI 4.0, systems engineering, process modeling.**

I. INTRODUCTION

The term "Industry 4.0" is describing the upcoming changes in manufacturing systems, away from product-orientation towards service-orientation [1]. These changes are mainly caused by technology-driven approaches like cyber-physical systems (CPS) or new technologies in digitization and networking. The goal is to optimize production processes or create new business models. Hence, the future value creation process is composed by traditional manufacturing methods combined with high technological systems. This results in a complex cooperation of many different tools based on dynamical changing technologies. To handle this complexity is a major task in developing new industrial systems.

The first step to approach this problem is to structure a whole industrial system and elaborate its context. Trying to solve this objective, several leading German industrial associations introduced the Reference Architecture Model Industrie 4.0 (RAMI 4.0) [2]. The three-dimensional model is the focus of current research projects. Specialized for the application in the manufacturing area, the reference architecture is based on standards tailored to the industrial domain. The usage of RAMI 4.0 itself and its applied

methods are specified in the German standard DIN SPEC 91345 [3]. However, since the just mentioned specifications, presented in [4], and the underlying norm are only a theoretical concept, suitable applications using its methods need to be provided in order to consolidate its utilization. Previously, during the development of a domain-specific language (DSL) for Industry 4.0 [5], a major issue about missing formalization for modeling systems with the RAMI Toolbox has been pointed out. As the name assumes, this refers to a comprehensive set of tools designed for developing industrial systems based on RAMI 4.0.

In this paper, the mentioned issue is addressed by using well-known standards and engineering methods like model-driven architecture (MDA) or the ISO 15288. Uniting their concepts with those coming from RAMI 4.0 results in a more complete architecture description for industrial systems. Through the implementation of the aforementioned specifications into the RAMI Toolbox, additional instruments for the modeling of details within such a system are provided. However, for dealing with the complexity coming along with these new possibilities, a model-based development process for RAMI 4.0-based architectures is mainly proposed in this contribution. This is the next big step towards establishing and finding industrial acceptance for utilizing this approach. To evaluate every aspect addressed by a System of Systems (SoS), as the likes of one built on base of RAMI 4.0, a real-world Use Case characterizing a subway track manufacturing company is applied. This case study is provided by an external stakeholder and therefore allows deep insights into modern industrial systems, where system components and requirements can be derived and modeled according to real-world samples.

Therefore, the rest of this paper is structured as follows: in Section II, the concepts of RAMI 4.0 are analyzed towards their possibility to decompose a complex industrial SoS, which represents the basis for the overall development process. Furthermore, used standards and technologies are explained in detail. Afterwards, in Section III, the approach of the development process itself is provided. In Section IV, the industrial Use Case is modeled according to this process and further details on the application of the reference architecture are given. Finally, the contribution is concluded in Section V and the paper is summarized.

II. RELATED WORK

As motivated in Section I the goal of the presented approach is to define a specific development process for creating models of industrial systems. By doing so, RAMI 4.0 is used as architectural base, since it is specifically targeted towards modeling these kinds of applications. This

modeling process is supported by utilizing the previously described DSL and the concepts of the RAMI Toolbox. Moreover, common and well-known standards from the area of systems engineering describing its life-cycle and architecture are integrated within this approach. Therefore, according to the focus of this paper, this section briefly introduces standardized concepts that contribute to this goal.

A. Din Spec 91345

As visualized in Fig. 1, RAMI 4.0's goal is to enable the consideration of an industrial system based on domain-specific viewpoints. Doing so, the architecture itself is structured in "Life Cycle & Value Stream", "Hierarchy Levels" and "Interoperability Layers". On each of these axes, a different proportion of the system is treated, introduced in the following:

The horizontal axis of RAMI 4.0 deals with the different states an asset may have during its time of usage. This leads back to the fact that the future manufacturing process tends to migrate into an interlinked value creation network, where information is exchanged throughout the whole process, instead of communicating linearly. The integration of the standard IEC 62890 [6] helps providing a classification of a component over four different phases. Hence, the separation of type and instance are of importance in order to collect all data referring to the component. Furthermore, the standard specifies a second layer of abstraction, where a distinction between development and usage is described. The aim is integrated engineering of a system's component from its idea to its disposal, without losing data or information and the possibility to provide those at any time or state [7].

The vertical integration within a factory is represented by the Hierarchy Levels. Giving a rough guideline on how to classify components according to its application area, the two standards IEC 62264 [8] and IEC 61512 [9] are used for structuring. Derived from the well-known automation pyramid, 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). However, this structure only provides the base for a functional assignment, the actual implementation needs to be adapted for every system separately.
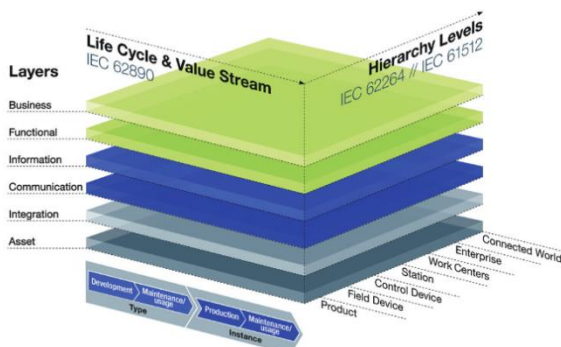

Fig. 1. Reference architecture model industrie 4.0 (RAMI 4.0) [10].

Finally, the top-down arrangement of the layers enables the structuring of each component within the system according to its features, appropriated to every viewpoint. For example, the Function Layer deals with services provided by the component, whereas the Information Layer holds its data. Furthermore, the Component Layer implements the physical viewpoint and therefore enables the real-world representation of the component.

B. System Life-Cycle Process

The development of a system needs to be structured according to the phases of its life-cycle in order to prevent confusion, misunderstanding or even conflict, according to Lake [11]. He defines four different kinds, each one dealing with another aspect of the system, namely Acquisition, Project, Development and Product. Derived from this, the life-cycle of systems development is pointed out as the most important task for engineering a system. During its utilization, it combines results and aspects of determined life-cycles for optimized decision-making and defining technical efforts. Hence, for providing information to decision makers, several sub-processes have been introduced, reaching from designing the system, followed by its development and evaluation.
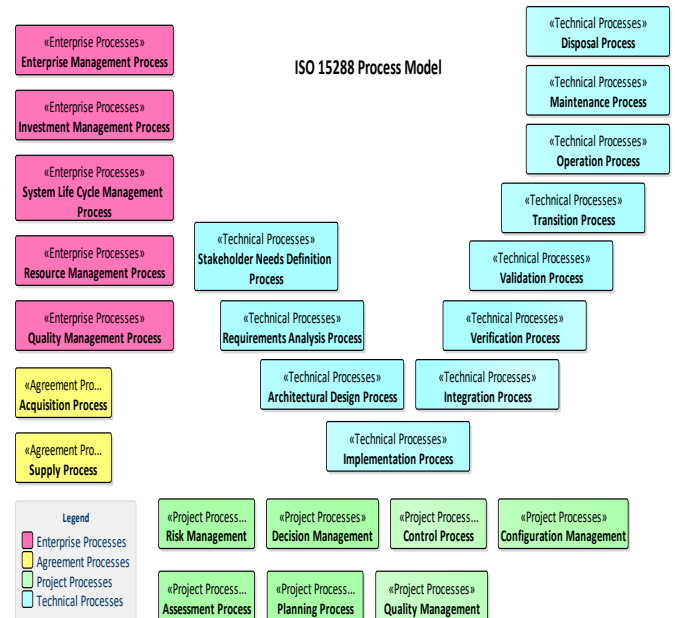

Fig. 2. ISO 15288 system life-cycle process.

With regard to this early approach, todays widely used standard ISO 15288 [12] is the main outcome resulting from the findings of Lake's development process, which makes use of the original introduced spiral model. Through slightly adopted phases and the integration of a V-model, this standard introduces an optimized process for the development of a system with the ultimate goal of achieving customer satisfaction. One big advantage of this framework is the possibility of adapting it to individual application areas by making use of only those parts needed for developing a specific system. Furthermore, by introducing the V-model, there is a process for evaluating every part of the system created as a result of executing the so-called technical processes on different abstraction levels of the system. The structure of the different stages and their respective processes is visualized in Fig. 2.

III. APPROACH

As outlined in [13] it is challenging to handle the complexity of distributed systems, especially if they provide critical infrastructures, as it is the case for Industry 4.0. A broadly excepted approach to deal with this complexity during the engineering process is the concept of Model Driven Engineering (MDE), which serves as an umbrella term for model-based approaches. Derived from this, a common method for developing systems in specific domains is introduced in [14]. The so-called Domain Specific Systems Engineering (DSSE)-Approach is split up into three major phases during the development process. In the first step the System Analysis defines stakeholders, requirements and system boundaries. Following, as result of the System Architecture, used technologies and components of the system are described. This serves as base for the detailed Design & Implementation Phase, where the generation of machine-readable code according to the defined model takes place. As previously mentioned, by decomposing the architecture, the DSSE-Approach can be described in a more detailed way by integrating the concepts of the ISO 15288. According to this, the following phases are decomposed to the technical processes of the standard to describe the development of a system in more detail:

1) Business Analysis Process
2) Stakeholder Needs Definition Process
3) Requirements Analysis Process
4) Architectural Design Process
5) Design Definition Process
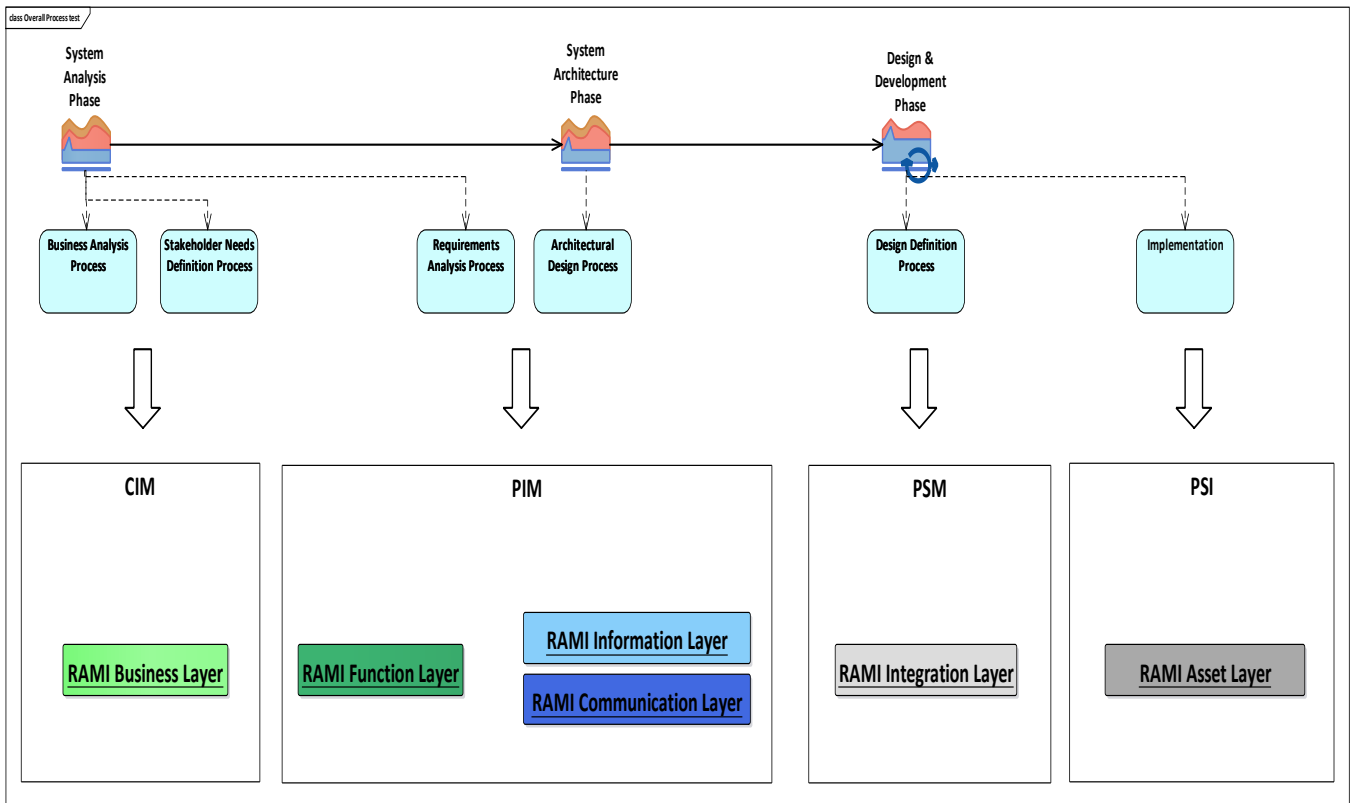6) Implementation

Fig. 3. Process model for developing RAMI 4.0 applications.

Additionally, every single of these processes includes further engineering tasks for developing a specific part of the system. As an example, the requirements specification is part of the Stakeholder Needs Definition Process, whereas the Architectural Design Process includes tasks for developing the information or communication architecture. Each task satisfies a goal, interest or desire coming from a stakeholder. Therefore, every stakeholder interested in or affected by the system is included during the development of the system and in the architecture description according to ISO 42010 [15].

Considering an Industry 4.0 application as an interdisciplinary SoS, the architecture-focused concepts of Model Driven Architecture (MDA) [16] appear suitable to analyze, decompose and develop such an industrial system. However, other than the Model Driven Software Development (MDSD) approach, MDA primarily focuses on the structuring of specifications rather than on the generation of implementation artifacts. By doing so, MDA introduces different viewpoints and their relation referred to as model transformation. Hence, the previously defined processes deliver the single MDA viewpoints and views, which are described with models. In more detail, the System Analysis Phase with all its tasks delivers the Computational Independent Model (CIM), including models for describing business cases, processes and requirements in order to give an overview of the system without going into implementation. Consequently, a common understanding of the intended functionality is elaborated during the Platform Independent Model (PIM). The high-level architecture of the system including superordinate functionalities and generic actors are composed as a part of this viewpoint. The next step, the System Architecture Phase, deals with used technologies and exchanged information resulting in the Platform Specific Model (PSM), where the components of the system are described in detail too. Finally, the Platform Specific Implementation (PSI) and its artefacts are delivered by an

iterative execution of the Design & Development Phase. The mapping of the individual phases of the DSSE-Approach to the MDA viewpoints is illustrated in Fig. 3.

Since this process model is developed by utilizing the concepts of the Agile Design Science Research Methodology (ADSRM) [17], a flexible development method for creating the process model is available, where changes may take place in every process step. However, as specified by this method, a suitable case study needs to be available for specifying the boundaries and conditions for future development phases. Therefore, the next step makes use of the previously defined specifications and defines requirements for the development. Those requirements are applied to the case study by its practical implementation and evaluated afterwards to complete one iteration of the ADSRM cycle. Hence, the first iteration of this cycle resulted in the creation of a DSL for RAMI 4.0, as described in [5]. Consequently, the next iteration deals with the development of the process model outlined in this contribution. This is initiated by drawing up a new case study and specifying requirements adjusted to the methodology mentioned before. Additionally, the previously developed DSL is taken for use in order to model the case study according to the development process and enable its evaluation.

A. Case Study Design and Requirements

According to ADSRM, the first step is to draw up a suitable case study. In this case a typical use case concerning Industry 4.0 is presented. More precisely, this example makes use of a subway track manufacturer that provides their customers a possibility for the creation of individual subway tracks according to their needs. This means, business models and requirements as well as system components are derived from real implementations recreating an actual industrial system. The manufacturer implies all tools used for customer interaction as well as the factories where the subway tracks are produced. This is based on the ideology of Industry 4.0 inheriting the goal of fully automated production processing, like handling the order or mechanically producing the subway track itself. This means, all machines should communicate with each other in order to find the optimal solution concerning resources. However, to not exceed the scope of this work, only a modest number of elements are used to fulfill the specified requirements. In this case, for elaborating a development process for Industry 4.0 applications, the following requirements have been specified by using methods described in [18]:

1) The process should manage the continuous creation of an industrial system.
2) The process must consider the design principles divide and conquer as well as separation of concerns
3) If the manual workload exceeds, parts of the process or the process itself can be automated.

IV. APPLICATION OF THE PROCESS MODEL

In the following, the case study involving the development process for industrial applications will be outlined. By doing so, the application context for the MDA-approach described in the previous section and design decisions resulting from each phase of the development process are explained. Finally, the concepts and experiences from this case study will be discussed towards their applicability in order to evolve a common development approach for Industry 4.0.

A. Case Study Modeling

According to the considerations mentioned before, the modeling of the case study example follows the steps of the MDA-based development process proposed in Section III. However, for modeling this system in the context of RAMI 4.0, a specifically tailored DSL is applied. This DSL utilizes the RAMI 4.0 structure and is implemented as an UML-based profile, consisting of functional or structural components and their relations [5]. Furthermore, as mentioned before, this case study is designed with the help of the RAMI Toolbox (https://rami-toolbox.org/) due to its comprehensive features.

The first step of the process model is realized by the Business Analysis Process. This is the task of defining business actors with their respective goals and the enclosing system boundaries. In this example the Business Layer contains three major actors, the customer, the manufacturer and the raw material supplier. Each of these actors is connected to one or more Business Use Cases (BUC), representing the interconnection and executed business processes between them. However, since RAMI 4.0 defines several value-creation chains [19], the BUCs can be aligned according to their objectives they try to achieve. For example, the BUC "Produce Subway Track" is part of the value-creation chain "Production", whereas "Design Subway Track" belongs to "Product Development". Subsequently, the system requirements are elaborated. This is done during the Stakeholder Needs Definition Process. Therefore, a single BUC is decomposed into several High-Level Use Cases (HLUCs), specifying the realization of functions performed by the BUC. During this process it is important to consider the business goals of each actor, which have been previously defined. One HLUC therefore builds the base for one or more requirements the system must fulfil. Those requirements are defined by evaluating the intended functionality of these use cases or they originate from a requirement engineer talking to architecture stakeholders. However, these two phases combined deliver the CIM as introduced by MDA.

In the next phase, the Requirements Analysis Process, the previously specified requirements are taken for use in order to elaborate the single components of the system. This process is called FAS Methodology and has been provided by [20]. Summarized, one Use Case realizes each requirement by describing it with an Activity Diagram. Consecutively, one or more activities can be grouped together in regard to their intended purpose. This means, each of these functional groups represents one component of the system. However, depicting the Logical Architecture, those components are represented by Logical Actors. The interconnection between the actors is realized by Use Cases, describing the functionality of the system. Concerning MDA, the relation between Business and Function Layer of RAMI 4.0 is realized by a model transformation, tracing Business Actors into Logical Actors as well as describing requirements with Use Cases.

After the functional architecture is specified, the Architectural Design Process is executed. More precisely, the Information and Communication Layers are used to describe

which data is needed by the respective functions and how it is exchanged. Therefore, firstly it must be specified which Logical Actor contains what data. Furthermore, in order to process the data correctly, the input and output of the executed functions need to be defined. A typical example for Industry 4.0 would be the automatic processing of a product order. After receiving the order, its data needs to be available for involved machines to decide whether it is possible to manufacture the desired product with the available resources or if the order has to be delayed or even declined. Therefore, the required data needs to be provided by a system component being accessible by other manufacturing components. In order to process this data properly, those system components have to be connected with each other, which is specified in the Communication Layer. Therefore, the type of connections between the single components is specified by modeling the interfaces, which are realized as ports. Since RAMI 4.0 is described as a service-oriented architecture (SOA), every component that needs data includes a request point whereas every component that provides data includes a service point. Summarized, those three layers resulting from the two intermediate phases of the development process build the PIM of MDA.

After the architectural composition of the system is done, the Design Definition Process delivers the Integration Layer and its MDA's adequate, the PSM. As the name assumes, this viewpoint provides a detailed view on the system. By doing so, the system components are modeled based on real-world elements containing as much details as necessary and including all subcomponents to provide information on each abstraction level. For example, the subway track contains several elements like different sensors, some wiring, a control unit and a condition monitoring unit. Those units itself consist of various modules or connecting elements, as seen in Fig. 4. However, the respective subcomponents are defined by applying the FAS methodology to already specified higher-level system components. In this way, the different abstraction levels can be considered by modeling the system.

Finally, the Implementation deals with the actual application of the system and its components. Suitable approaches for this case would be the utilization of AutomationML and OPC UA. As described in [21], models regarding these methodologies can be transformed into files containing engineering data or specifications for exchanging this data between the elements. Hence, to fulfil the concept of round-trip engineering, a possibility to update the model from real operational data needs to be provided as well. However, this would exceed the scope of this work and has to be considered in future projects.

B. Findings

Although this generic example enabled the evaluation of the development process on a superficial perspective, the used concepts worked well in general. However, the next iteration step of ADSRM needs to deal with more specific problems. For example, it was shown that the general process and the integration of the ISO 15288 deal well with structuring the whole system according to architectural descriptions and managing its development cycle. However, when it comes to developing detailed views of the system, the

coarse granularity of the process model bars the way to a fully devised system description. Hence, this process needs to be refined by including more detailed subordinate tasks that need to be fulfilled in each of its phases. The problem of the superficial definition can be attributed to the specification of RAMI 4.0. Although this reference architecture is already established in the German standards, currently only the frame to work in has been elaborated. In contrast to the Smart Grid Architecture Model (SGAM), which has been in the focus of development and research projects for almost a decade from today's point of view, advances in the industrial domain are just at the starting point [13]. Therefore, an architecture framework based on the ISO/IEC 42010 needs to be provided, which formalizes RAMI 4.0 in a more specific way, for example by deriving viewpoints and views adjusted to each layer. Suitable approaches providing a solution for this problem may be the Unified Architecture Framework (UAF), the Reference Model of Open Distributed Processing (RM-ODP) or the Software Platform for Embedded Systems (SPES), amongst others.
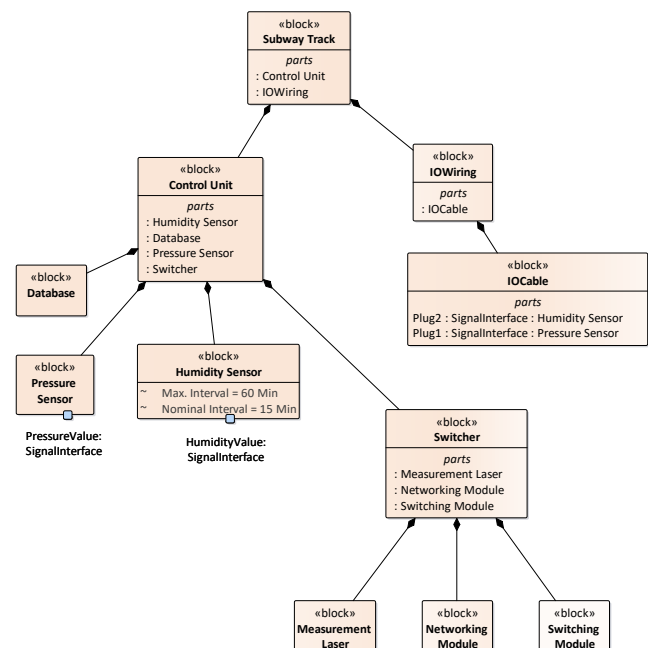

Fig. 4. SysML block definition diagram of subway track.

V. CONCLUSION AND FUTURE WORK

The work proposed in this paper points out the need for enhanced methods, such as MDA, in order to gain a holistic view on systems based on Industry 4.0. The technical framework to develop and describe such a system has been introduced with the standardization of RAMI 4.0. However, since this is just an approach existing in theory, the practical applicability needs to be ensured in the first step. Concerning this, previously a DSL giving the possibility to develop such an industrial system has been developed. In further consequence, a common development process dealing with the consistent creation of these systems needs to be provided. Therefore, this paper contributes the realization of a process model adjusted to RAMI 4.0 by integrating well-known standards like the ISO 15288 or MDA. This process is utilized to demonstrate the applicability in context of a simplified demonstration example derived from a real-world

case study.

However, even if the presented concepts demonstrate a possible way on how to emphasize a common development process in regard to RAMI 4.0, they do not claim to be a "ready to use" methodology. This can rather be seen as a first step on the way to establishment. One of the most important tasks going further on this way will be the more detailed decomposition of RAMI 4.0 for developing specific industrial architectures, as explained in Section IV. After achieving this, several other steps can be executed built on the basis of this architecture definition. For example, the DSL needs to be adapted in order to describe more detailed views of a system. This is done by integrating additional technologies like the Business Process Model Notation (BPMN) in the Business Layer or SoaML for describing communication structures. Based on these enhancements, the process model proposed in this contribution can be refined as well by defining more specific sub-processes in each phase. To achieve this, it will be necessary to integrate the knowledge of domain experts from every viewpoint the architecture complies in order to derive a commonly accepted decomposition. This has to be done with the help of a more sophisticated case study and external stakeholders during the execution of ADSRM's next iteration step.

REFERENCES

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffman, *Industry 4.0, Business and Information Systems Engineering*, Springer, 2014, pp. 239-242.

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

[3] *Reference Architecture Model Industrie 4.0*, DIN Standard 91345-2016.

[4] R. Heidel, M. Hankel, U. Döbrich, and M. Hoffmeister, *Basiswissen RAMI 4.0: Referenzarchitekturmodell und Industrie 4.0-Komponente*, Beuth Verlag, 2017.

[5] C. Binder, C. Neureiter, G. Lastro, M. Uslar, and P. Lieber, "Towards a standards-based domain specific language for industry 4.0 architectures," *Complex System Design & Management*, 2018.

[6] *Life-cycle Management for Systems and Products Used in Industrial-process Measurement, Control and Automation*, IEC Standard 62890-2016.

[7] J. Wollert, "Die nächste generation der automatisierungstechnik: Cloud-technologie als schlüssel," *Elektronik.De*, 2017.

[8] *Enterprise-Control System Integration*, IEC Standard 62264-2016.

[9] *Batch Control*, IEC Standard 61512-2001.

[10] *Umsetzungsstrategie Industrie 4.0: Ergebnisbericht der Plattform Industrie 4.0*, Bitkom, VDMA, ZVEI, 2016.

[11] G. Lake, "Thoughts about life cycle phases," prested at INCOSE International Symposium, 1997.

[12] *Systems and Software Engineering - System Life Cycle Processes*, ISO/IEC/IEEE Standard 15288-2015.

[13] L. J. Kirsch, "The management of complex tasks in organizations: Controlling the systems development process," *Organization Science*, INFORMS, 1996, pp. 1-21.

[14] C. Neureiter, *A Domain-Specific, Model Driven Engineering Approach for Systems Engineering in the Smart Grid*, MBSE4U - Tim Weilkiens, 2017.

[15] *Systems and Software Engineering - Architecture Description*, ISO/IEC/IEEE Standard 42010-2011.

[16] R. Soley, "Model driven architecture," *OMG White Paper*, 2000.

[17] K. Conboy, R. Gleasure, and E. Cullina, "Agile design science research," in *Proc. International Conference on Design Science Research in Information Systems*, 2015.

[18] K. Pohl and C. Rupp, *Basiswissen Requirements Engineering*, 2015.

[19] VDI/VDE-Gesellschaft, *Wertschöpfungsketten in Industrie 4.0*, VDI, 2014.

[20] T. Weilkiens, J. G. Lamm, S. Roth, and M. Walker, *Model-Based System Architecture*, John Wiley & Sons, 2015.

[21] L. Berardinelli, S. Biffl, A. Lüder, E. Mätzler, T. Mayerhofer, M. Wimmer, and S. Wolny, "Cross-disciplinary engineering with AUTOMATIONML and SysML,"*Automatisierungstechnik 64*, 2016.

Christoph Binder obtained both his BSc. and his MSc. in computer science from the Salzburg University of Applied Sciences. Currently, he is a junior researcher and lecturer in the Center for Secure Energy Informatics in Salzburg, where he is extending his work of a "standards-based domain specific language for industry 4.0 architectures" towards a PhD. His main research interests include systems engineering, architecture development and evaluation as well as process optimization, especially in the field of industry 4.0.

Christian Neureiter received his PhD degree in computer science from the Carl von Ossietzky University, Oldenburg, Germany. He is an associate professor in computer science and energy informatics at the Salzburg University of Applied Sciences where he is head of the Domain Specific Systems Engineering (DSSE) research team in the Center for Secure Energy Informatics. His main research interests include, amongst others, domain specific and model driven systems engineering in several application areas like smart grid, smart cities, automotive and industry 4.0.

Goran Lastro obtained both his BSc. and his MSc. in computer science from the Salzburg University of Applied Sciences. He is a lecturer at the Salzburg University of Applied Sciences and researcher at the Center for Secure Energy Informatics, where he is a lead developer in the Domain Specific Systems Engineering (DSSE) research team. His main research interests include systems engineering, architecture development, architecture evaluation and process optimization in multiple domains such as the smart grid and industry 4.0.