Towards a Model-Centric Approach for developing Functional Architectures in Industry 4.0 Systems

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*—Nowadays popular term "Industry 4.0" characterizes a currently occurring industrial revolution, promoted by advances in research and development. For example, so-called Cyber-physical Systems (CPS) offer new automation possibilities tailored to industrial manufacturing, resulting in constantly increasing complexity. In particular, the link between requirements engineering and constructing the technical architecture faces a major gap when developing future industrial systems. 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. However, at the current point of view, only the frame to work in is specified by this already standardized framework, as it is missing formalizations and common methods for describing such a system in detail. Thus, this paper introduces an approach for developing the functional architecture of an industrial system on the RAMI Function Layer with special regard to close the gap between requirements and technical components. To achieve this, the concepts of RAMI 4.0 are analyzed and well-known model-based methodologies for developing system architectures are applied. In the end, the application of the approach itself is demonstrated by utilizing a real-world case study.**

***Index Terms*—Functional Architecture, Traceability, Industry 4.0, Reference Architecture Model Industrie 4.0 (RAMI 4.0), Domain-specific Systems Engineering (DSSE)**

I. Introduction

Technological advances promoted by the Internet of Things (IoT) lead the path to a new kind of value creation in contemporary manufacturing companies. This trend, nowadays widely known by the term "Industry 4.0", offers those global players as well as local companies new possibilities of pursuing novel business models with the goal to optimize production by reducing expenses at the same time. These changes contribute to technology-driven approaches resulting in a new form of automation driven industry, which drifts away from the original product-orientation towards technology-oriented services. One of the main outcome supporting this transformation is the emergence of Cyber-physical Systems (CPSs) [1]. Those IoT based units are mainly intelligent components of a manufacturing process, which are able to find the economically most valuable decision on their own. Consequently, the interconnection of multiple CPSs forms a service-oriented value creation network facing challenges required for the application of Industry 4.0, like self-sustaining production or real-time information handling [2].

However, accompanied with those new opportunities a new level of complexity is approaching. Applying the classification scheme proposed in [3], the change can be explained in more detail. By clustering the attributes *dynamic* and *alterability* as well as *diversity*, *variety* and *scale*, a two-dimensional classification chart distinguishing between four kinds of systems is provided. As introduced by this scheme, a simple system is comprised of few elements which are statically interconnected. However, adding a large number of homogeneous elements or a dynamic interaction behavior emerges a complicated system. Including both of these characteristics results in the definition of a complex system. According to this scheme, a traditional manufacturing system can be classified as a complicated system, whereas an Industry 4.0 based system is considered to be a complex system. Furthermore, with the fact that CPSs being systems themselves, the term System of Systems (SoS) is suggested to be used in order to emphasize the autonomous character of its individual participants. This is mainly supported by utilizing the traits *independent operation*, *geographic distribution* and *evolutionary behavior* [4].

One of the main challenges promoted by this complexity is the dissolving of the clear structures between the system functions and the technical components. In traditional manufacturing systems, one function is realized by one physical component. Nowadays, through the introduction of CPSs, one function is distributed over many physical elements, and one element offers more than one function. Since functions are usually developed to fulfil certain requirements, the link between the physical parts and the functional requirements is fading. This makes it difficult to define the importance and exchangeability of each component. Having recognized this issue, 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], which is standardized in the DIN SPEC 91345 [6]. In order to ensure the application of this architectural framework, the authors of this paper previously introduced a Domain Specific Language (DSL) [7] as well as an associated development process [8], which are making use of common Model Based Systems Engineering (MBSE) methods. However, the main findings of both of these works are the missing specifications of RAMI itself. This impedes the detailed development of future industrial systems concerning the mentioned aspects at the current point of view.

Therefore, this paper introduces two major contributions. First, the refinement of the RAMI Function Layer is introduced by specifying viewpoints and model kinds in order to support the application of MBSE. To increase the effectiveness of this approach, domain-specific elements are implemented and the utilization of Model Driven Architecture (MDA) is demonstrated. The second part describes the evaluation of the present approach by applying a real industrial case study, which is provided by a domain-expert. However, the goal of this approach is not to reinvent the wheel, but to use well-known and widely applicable technologies and standards. Thus, the presented methodology makes use of adapting the architecture's viewpoints to the ISO 42010 or implementing SysML or the FAS Methodology for providing established methods tailored to functional aspects in systems engineering. This not only takes the application of RAMI 4.0 one step further but also cares about gaining acceptance in the community, which is important for turning into a common basis for developing future industrial systems.

To address these aspects, this contribution is structured as following: In Section II an overview of RAMI 4.0, the ISO 42010 and several methods for developing functional architectures is given. Hereafter, the used approach to challenge the problem is stated in Section III. The implementation of the mentioned aspects into the RAMI Toolbox is described 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. In more detail, the horizontal axis of RAMI 4.0 deals with the different states an asset may have during its time of usage. By falling back to the criteria introduced in the standard IEC 62890 [9], the aim is collecting data referring to the component throughout its 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. In the second axis, the vertical integration within a factory is represented by the Hierarchy Levels. Based on IEC 62264 [10] and IEC 61512 [11], 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). Finally, the top-down arrangement of the layers enables the structuring of the system according to the feature of its components across six viewpoints. Thus, the Business Layer defines processes and boundaries of the system, resulting in the elaboration of requirements. Those requirements build the base for the future development of the system, in particular the specification of services displayed on the Function Layer. The Information Layer deals with handling all kind of data, whereas the Communication Layer contains connections and interfaces within the system components. Following this principle, the Integration Layer enables the digitalization of components by specifying Human-machine Interfaces (HMIs). At last, the Component Layer implements the physical viewpoint and therefore enables the real-world representation of the component.

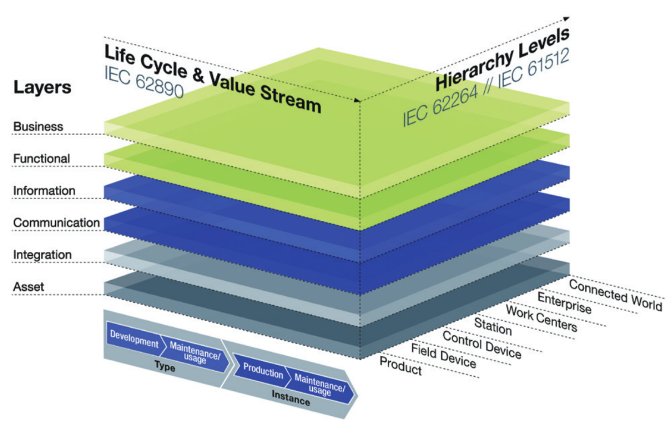

Fig. 1. Reference Architecture Model Industrie 4.0 (RAMI 4.0) [12]

B. ISO 42010

Standardized approaches for the creation, analysis and maintainability as well as for the improvement of the communication between different stakeholders are needed due to the growing complexity in systems. The ISO/IEC/IEEE 42010 International Standard [13] provides such 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. Therefore, the standard defines all information an architecture framework shall include, explained in the following:

- Information identifying the architecture framework
- Identification of one or more stakeholders
- Identification of one or more stakeholder concerns
- One or more model kinds to describe the viewpoints
- Correspondence rules between the viewpoints

Besides the specification of architecture frameworks and its viewpoints, Architecture Description Languages (ADLs) are also one core element of 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 SysML. Another very important artefact, when it comes to modeling, is a well defined process. Such a process, like MDA introduced by the Object Management Group (OMG), should support the task of modeling. Process steps provide information on how to model the system when using the framework and the developed DSL.

C. Functional Architecture for Systems

Due to a lack in common approaches for developing functional architectures, especially in the context of MBSE, the desire for such a method has become more and more obvious. Thus, the Functional Architecture for Systems (FAS) method has been introduced in [14]. It provides a methodology for developing a technology-independent, function-oriented description of the system as a block-oriented structure. The main reason this method needs to be applied in modern systems engineering is the upcoming complexity. More precisely, in complex systems usually a function is deployed on many physical components and a physical component realizes more than one function. Therefore, the functional architecture can be interpreteted as an interface between the requirements and the physical architecture. By doing so, the FAS provides three main modeling elements:

- Functional Element: This is described as an abstract system element that defines a relation between at least one input and at least one output by means of a function.
- Functional Group: In terms of FAS, this is a set of strongly related use case activities.
- Functional Interface: As the name assumes this element defines a set of inputs and outputs of a Functional Element.

Although the FAS method is independent of any modeling language, it is recommended to use SysML for its implementation due to integration opportunities and its acceptance by the community. By doing so, this methodology follows a simple process. At first, the behavior of all functional requirements is described by use cases and its activity diagrams. Subsequently, all strongly related activities are summarized into Functional Groups, which could also contain actions or not refined functional requirements. Having elaborated all grouped activities, the next step is to trace each Functional Group into one Functional Element, which build the base for developing the functional architecture. However, by utilizing SysML the interconnection of these elements as well as their interfaces can be displayed with a Block Definition or an Internal Block Diagram.

III. Approach

As already mentioned, the goal of this approach is to refine the RAMI Function Layer in order to provide an interface between the requirements definition and the technical architecture. Since managing the accompanied complexity in SoS is not a completely new topic, most contemporary Model-driven Engineering (MDE) approaches contain some kind of methodology for dealing with this. For example, Domain Specific Systems Engineering (DSSE) introduces three major phases for developing a system, which are refined in more detail by applying the ISO 15288 [15]. As outlined in [16], a detailed description of the system's functions needs to be performed in order to set up the specifications for the system architecture. This process is supported by utilizing a top-down development process of the modeling paradigm MDA. Taking this into further consideration, the approach presented in this paper needs to make use of the Computation Independent Model (CIM) in order to specify the requirements. Further, these are utilized for developing the single system components in the Platform Independent Model (PIM). Mapping this to RAMI 4.0, the Function Layer has to implement model kinds for refining the requirements, developing the functional architecture and managing the link between requirements and physical parts of the system.

Due to the unpredictability of future applications affected by the high rate of change in terms of used methods and the dimension of projects, an agile approach needs to be applied for elaborating the details of the Function Layer. 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 [17].

A. Case Study Design

This specific example makes use of a real-world industrial use case, an manufacturer of car engines. Since corresponding information have been provided by a company partner, deep insights into single manufacturing processes and detailed part specifications were made available. This is especially important when it comes to meeting Industry 4.0 based specifications like the implementation of fully automated production processing or interconnecting all machines with each other. However, to not exceed the scope of this work, not the whole manufacturing system but only a clear defined aspect of it inheriting one specific business model is utilized to deal as base for the functional architecture. Hence, this example proposes the development of a combustor for burning the fuel together with compressed air in order to transmit the energy to the turbine. In order to fulfil the specifications of ADSRM, some requirements are derived from this use case, based on the methods introduced in [18]. Thus, the RAMI Function Layer should (1) meet the needs and concerns of domain-specific stakeholders, (2) deal as an interface between the requirements and the technical architecture and (3) consider the specifications of RAMI 4.0 as well as the indicated theoretical concepts in its defintion of the Functional Layer.

IV. Implementation

A. Viewpoints 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. Summarized, this results in the definition of the following views:

- FAS View
- Black Box View
- White Box View
- Actor Mapping View

B. Adaption of DSL

After defining the stakeholders concerns and the corresponding views, the already existing DSL for developing systems based on RAMI 4.0 needs to be adapted. This language contains all modeling elements for describing an industrial system on each of the six abstraction layers of RAMI 4.0. Thus, the single elements, derived from the Unified Modeling Language (UML), are assembled to build the metamodel, which serves as representation of the real world by formulating the dependencies between the elements. However, the first step of adapting this UML profile for providing the possibility to develop functional architectures is the definition of model kinds for each viewpoint. Since the method of choice for developing system functions from functional requirements is the FAS method, its corresponding viewpoint needs to implement all model kinds that contribute to this goal. Therefore, requirements are refined by Use Cases in Use Case Diagrams. Moreover, the Use Cases are further described by Activity Diagrams. Consequently, the elaborated functional requirements, activities and actions are summarized into Functional Groups in a particularly designated Function Development Diagram. This diagram type implements all DSL elements needed for applying the FAS method like Functional Groups or Functional Elements. Those modeling elements are additionally used to be traced into each other in an Actor Mapping Diagram. However, since the FAS related elements are derived from SysML, the generalization needs to be adapted in the UML profile as well in order to add all specific attributes used in SysML. In the next viewpoint, the Black Box Model, the dependencies between the single Functional Elements as well as their Functional Interfaces are developed. For this case, a SysML Block Definition Diagram or the aforementioned Function Development Diagram is utilized. Finally, in the White Box View, the so-called "chain of effects" is modeled with the help of a SysML Internal Block Diagram.

C. Tool-Support

The RAMI Toolbox[1] is a specifically designed Software with the aim to provide tool-support for developing system architectures based on RAMI 4.0. Available as an Add-In for the modeling software Enterprise Architect (EA), the toolbox loads the UML profile with all modeling elements and provides a tool-set containing different functionalities tailored to enhance the usability by automating repetitive modeling processes. Therefore, in order to support developing the functional architecture, some methods are added to the RAMI Toolbox. Since transforming all requirements within a large model is a time-consuming task, a user interface deals with collection all functional requirements and provides an overview of connected Use Cases. By doing so, connections can be modified and new Use Cases are created easily. Furthermore, modeling all activities and actions of each created Use Case by hand is monotonous and repetitive for the system architect. This task is improved by providing a so-called FAS-Wizard, which deals with the automation of this methodology. Additionally, some further functions implement the possibility to create the RAMI matrix and arranging Functional Elements to each plane according to their position attributes.

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

V. Application

In the following, an overlook of the modeled case study[2] is given. This example makes use of the specifications set in Section III and the methods described in Section IV. Since a real-world case study of a car engine manufacturer is utilized for creating the architecture model, business models, the system context and requirements are derived from real manufacturing systems actually applied in the industry. According to these considerations, the first step is to model the Business Layer of RAMI 4.0. Since the objective of this layer is to elaborate the functional requirements, suitable models for defining the system context and stakeholder goals have to be applied. However, describing this in more detail would exceed the scope of this work and is going to be proposed in another contribution. Therefore, this example makes use of the following functional requirements, summarized into requirement clusters: (1) produce lightweight parts, (2) monitor bleed air induction and (3) turbine movement. As observed from this context, those requirements are spread over different granularity levels of the system. The first requirement deals with the production line, the others with the system to produce itself. More precisely, the System of Interest (SoI) is modeled on abstraction layer 1 and beyond, whereas the supersystem is modeled on level 0. This is the point where the RAMI cube is missing specifications, because it provides only the formulation of one granularity level.

In order to provide a puncture through the modeled case study, the requirement group turbine movement is used for further explanation. Thus, in the first viewpoint of the Function Layer, the FAS method is applied for developing Functional Elements from the requirements. To do so, a functional requirement is refined by a primary Use Case. The Activity Diagram of this Use Case depicts the behavior of the requirement by describing the sequence of events, as depicted in figure 2. The displayed image indicates the technical description of how to actuate the turbine. The process is triggered by inducting the inlet air into the engine. Hence, the goal of the next action is to compress this air in order to increase its density. Subsequently, fuel is added to the engine, which is burnt with the help of the compressed air. This combustion process releases heat, which increases the total energy of the fluid and therefore is able to actuate the turbine, hereby called output work. However, the side effects of this movement process is the release of exhaust on the one hand and some internal work on the other hand. Consequently, summarizing those Actions and other ones from Activity Diagrams not explained as well as non-functional requirements results in the definition of Functional Groups. Thus, in this example, the groups *compress air*, *burn fuel* and *move turbine* are resulting from these associations. Now, those functions can be described in more detail. This is done by modeling it as a black-box in the corresponding viewpoint and supported by the previously described DSL elements.

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

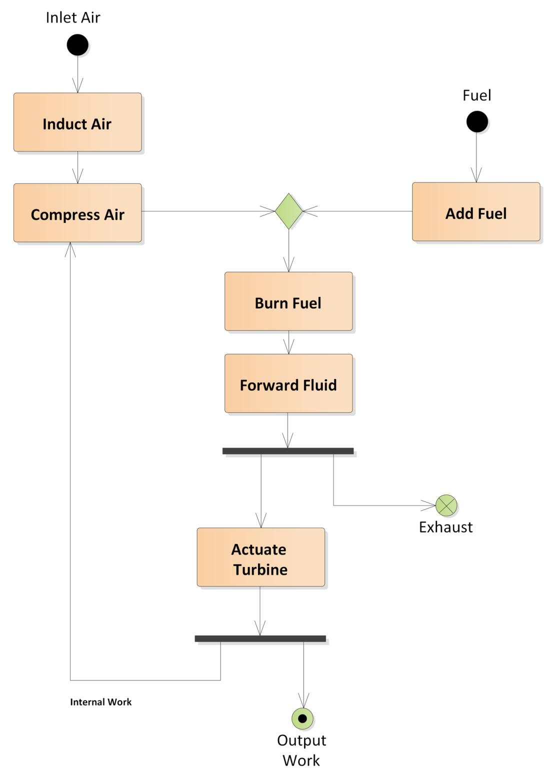

Fig. 2. Activity Diagram delineating the turbine movement

The function itself is depicted as a SysML block with interfaces defining input, output, disturbance and interference. Therefore, the input of the function move turbine is the heated fluid, whereas the outputs are internal work, output work and exhaust. An example for an interference would be some mechanical problems while a disturbance could be too less friction for the turbine to move fluently or too less heat from the burnt fuel. Next, the white-box representation of this SysML block is modeled with an Internal Block Diagram in the White Box view. More precisely, the hot fluid stream causes the turbine to rotate, which also rotates the output shaft. Furthermore, some energy flows back in order to move the compressor. In the last step, the exhaust is emitted by particular chambers. The depiction of the black-box and the white-box model is visualized in figure 3. In the last viewpoint, the Actor Mapping Model, the traceability between the requirements, the Functional Groups and the Functional Elements are depicted. According to this case study, the requirement *turbine movement* is traced to the Functional Group *move turbine*, which traces to the Functional Element *turbine*. From this point on, the turbine is considered as a part of the system and the interconnection as well as the technical description can be modeled on the underneath layers of RAMI 4.0. However, after modeling the SoI, more specific the parts to manufacture, on the granularity level 1 and beyond, the corresponding supersystem can be considered on the top level. Since the process for developing this level is the same as mentioned above, only a short overview is given.

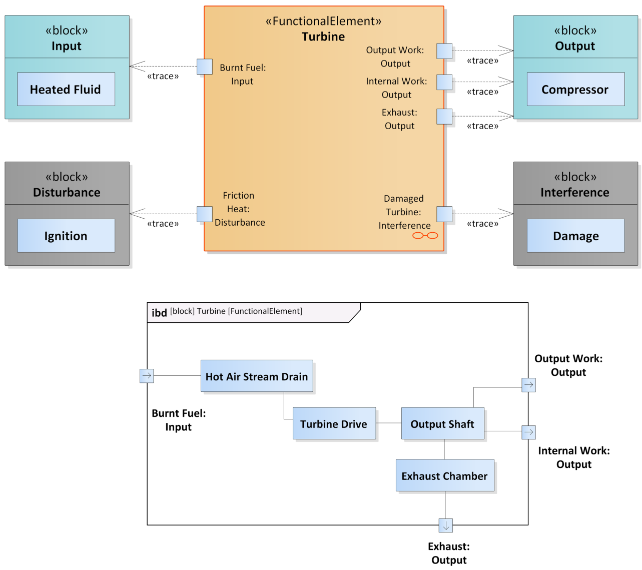

Fig. 3. Black- and white-box perspective of Functional Element

Thus, first the requirements for are derived from the ones specified for the single parts of the system and new ones are elaborated in order to create the production line. Then, by applying the FAS method, the single manufacturing processes are described with use cases and Activity Diagrams in order to refine the requirements. Consequently, all needed machines, raw materials, production planning and transport routes can be defined by summarizing the single actions and depicting them as Functional Elements. From this point on, after visualizing the black- and white-box perspective, Industry 4.0 attributes and interconnections can be added to the system components.

VI. Conclusion & Future Work

The approach proposed in this paper outlines the need for advanced methods when developing current and future industrial systems, in order to deal with the complexity that comes with advanced technologies CPSs are making use of. Since MBSE offers a lot of features tailored to deal with this complexity, it should be the method of choice. Thus, with the introduction of the technical framework RAMI 4.0 as well as the corresponding DSL and a development process, the first step towards developing current and future industrial systems has been set. However, with all introduced concepts being encountered only a superficial perspective, further refinement and specifications need to be formalized. Since there is a major gap between the requirements and the technical architecture, the first step to approach the issue of missing formalization is set by defining a viewpoint that servers as an interface connecting those different concerns. Thus, this paper proposes an approach for developing functional architectures of Industry 4.0 based systems by assigning system functions to technical components in order to fulfil the requirements. The result is thereby evaluated by an extensive real-world case study.
The outcome of this work can contribute to a lot of different follow-up projects. For example, after developing system components carrying out system functions, a detailed description of the technical architecture can be executed on the bottom layers of RAMI 4.0. By doing so, the metamodel including all domain-specific elements needs to be extended by the characteristics provided by SysML in order to enable a more specific representation. Furthermore, for describing a system on multiple granularity layers, the RAMI cube is missing an additional dimension. Therefore, the concepts provided by Software Platform Embedded Systems (SPES) [19] can be aligned in order to deal with this issue. These further refinements have to be done with the aid of ancillary case studies on the next iterations 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.

References

[1] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Design Automation Conference*. IEEE, 2010, pp. 731–736.

[2] 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.

[3] R. Haberfellner, O. de Weck,, E. Fricke, and S. Vössner, *Systems Engineering - Grundlagen und Anwendung*. Orell Füssli, 2015.

[4] M. W. Maier, "Architecting principles for systems-of-systems," *Systems Engineering: The Journal of the International Council on Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

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

[6] DIN SPEC, "91345: 2016-04," *Reference Architecture Model Industrie 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] International Electrotechnical Commission, "IEC 62890: Life-cycle management for systems and products used in industrial-process measurement, control and automation," 2016.

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

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

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

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

[14] T. Weilkiens, J. G. Lamm, S. Roth, and M. Walker, *Model-based system architecture*. John Wiley & Sons, 2015.

[15] International Organization for Standardization, "ISO/IEC/IEEE 15288: Systems and software engineering – system life cycle processes," 2001.

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

[17] 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.

[18] K. Pohl and C. Rupp, *Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level*. dpunkt. verlag, 2015.

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