

Towards a Standards-Based Domain Specific Language for Industry 4.0 Architectures

Christoph Binder, Christian Neureiter, Goran Lastro, Mathias Uslar and Peter Lieber

Abstract Advances in research and development paved the way for a new revolution concerning industrial manufacturing, called Industry 4.0. Cyber-Physical Systems (CPS) contain methods for ubiquitous monitoring of information and synchronizing it with any other component on each hierarchical level participating in the value chain. Developing Industry 4.0 architectures for pointing out the structural cooperation of these Systems of systems (SoS) is a challenging task including a lot of different stakeholders. To bring together knowledge and experience, a common methodology is necessary. Regarding this, several German industrial associations created a suitable reference architecture, called Reference Architecture Model Industry 4.0 (RAMI 4.0). In this paper, a Domain Specific Systems Engineering approach using a Domain Specific Language (DSL) based on the results of this reference architecture is proposed and evaluated by a suitable case study.

1 Introduction

Optimized management of available resources in order to maximize profit and at the same time reducing costs and expenses is the main goal of most manufacturing companies. Results from research and development in the area of information

Christoph Binder, Christian Neureiter, Goran Lastro
Center for Secure Energy Informatics Salzburg University of Applied Sciences, Urstein Süd 1,
A-5412 Puch/Salzburg, Austria, e-mail: {christoph.binder, christian.neureiter, goran.lastro}@fh-salzburg.ac.at

Mathias Uslar
OFFIS – Institute for Information Technology, Escherweg 2, 26121 Oldenburg, Germany, e-mail:
mathias.uslar@offis.de

Peter Lieber
LieberLieber Software GmbH, Handelskai 340, Top 5, 1020 Wien, Austria, e-mail: peter.lieber@lieberlieber.com

technology (IT) offer new possibilities to support this goal, which drive change in the present industrial area and lead the path to a new form of an automation driven industry, the so-called Industry 4.0. An example of a technology resulting from this change is cyber-physical system (CPS). CPS are mainly intelligent components of a manufacturing process, where they take over a specific task. The main advantage regarding productivity is, that CPS are able to find the economically most valuable decision on their own, based on information provided by other CPS taking part in the system [6]. As discussed before, the aim of Industry 4.0 is to advance automation in manufacturing companies. In order to achieve this, data between machines, processes and lots need to be exchanged over direct communication structures. In [8], several documents concerning the definition of Industry 4.0 and its components have been analyzed and some criteria describing it have been formulated. According to [23], a Digital Twin of the physical production system needs to be realized. As the name assumes, this concept refers to the digital representation of a physical asset, containing its information throughout the whole life cycle. Since some lots like screws do not have any possibility to communicate, its Digital Twin helps collecting useful data and provides them to other CPS in the industrial system.

To describe the structure and behavior of a system, a system architecture needs to be defined. Therefore, a tailored architecture to describe Industry 4.0 based systems must be available, managing the complexity that comes with it. Furthermore, this type of architecture should (1) be able to deal with dynamic changes that occur in this area (2) simplify complexity in order to improve user experience and (3) increase productivity instead of generating overhead, according to [25]. To deal with this issue, several approaches like RAMI 4.0 [7] or IIRA [16] have been launched. Although both reference architectures provide a certain tool-set for developing concrete industrial system architectures, the goal and used methods differ from each other. A comparison between RAMI 4.0 and IIRA can be found in [9].

The definition of RAMI 4.0 as well as its use and methods are clearly defined in the German norm DIN SPEC 91345 [5]. However, the application of RAMI 4.0 is not properly formalized and there are no solutions yet existing dealing with this. This is a big issue to solve, because ensuring the applicability of RAMI 4.0 could take this approach a big step forward towards establishing it in the industry. A suitable technology needs to be determined in order to deal with this open gap. Since the aspects concerning Industry 4.0 are usually too complicated to be treated with a generic technology, a domain specific approach tailored to the application domain must be specified. Furthermore, another required action is to provide comprehensive tool support for developing architectures of systems based on RAMI 4.0.

To address these aspects, this contribution is structured as follows: In Section 2 an overview of RAMI 4.0 and domain specific systems engineering is given. Hereafter, the creation of the domain specific language (DSL) is stated in Section 3. Based on a suitable demonstration example, the applicability is demonstrated in Section 4. Finally, in Section 5 the paper is summarized and then the conclusion is given.

2 Related Work

2.1 Domain Specific Architecture Framework

The Reference Architecture Model Industrie 4.0 (RAMI 4.0), depicted in figure 1, has been developed by the Plattform Industrie 4.0, a union of leading German associations in the industrial area. The three-dimensional model, derived from the already established Smart Grid Architecture Model (SGAM) [5], has been developed to create a common understanding. To do so, it contains standards and use cases related to Industry 4.0. Due to the big influence on the German industry of its creators, the reference architecture encloses multiple industry sectors and its span ranges over the complete value chain. This allows the system to be seen as a whole to find connections and display tasks or sequences of events over the whole process as well as create the possibility of providing a detailed consideration of parts from the system [1].

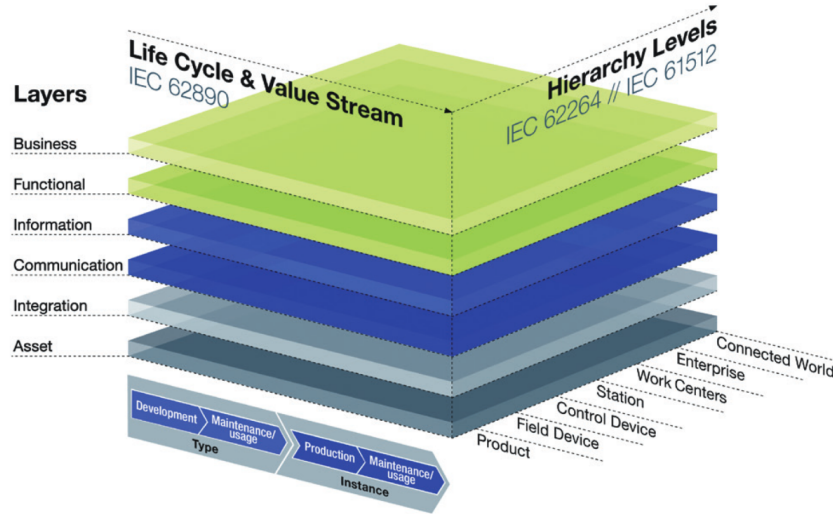


Fig. 1: Reference Architecture Model Industrie 4.0 (RAMI 4.0) [1]

To represent an asset over its whole life cycle, the horizontal axis has been introduced. It defines a product according to IEC 62890 [12] as type and instance, whereas the type represents the asset during development and prototype creation. However, the instance states a product as individuality and all its administration. Furthermore, the second axis deals with the classification of an item within the factory. Based on IEC 62264 [11] and IEC 61512 [10] a single product can be located regarding its spreading, from connected world over enterprise up to a single device

used in production [1]. The top-down arrangement of the layers enables the classification of subjects according to their task areas. This also enables the mapping of the single system development processes to their respective area. The system analysis takes place at the top layers, more detailed the Business Layer and the Function Layer. It describes the conditions and business processes the system has to follow and in further consequence the run-time environment of the system with all functionalities of its services and applications. The Information Layer provides all kind of data to its adjacent layers and the Communication Layer takes care of the connections within the system. To deal with all characteristics of CPS the Integration Layer has been defined only to display the physical objects on the Asset Layer.

2.2 Domain Specific Systems Engineering

Since Industry 4.0 is a widespread and challenging domain, engineering of systems is a complex task and needs to be confronted with suitable methods. Usually, in a complex field not a single system is constructed but an interaction of multiple homogeneous systems called System-of-Systems (SoS). According to [15], two disciplines need to be fulfilled. On the one hand, decent knowledge about the domain to operate with should be appropriated, on the other hand, it is mentioned that systems engineering management contributes significantly to the overall success. While a system is determined to fulfill a certain purpose, a SoS offers a solution for a more complex and extensive problem area [2]. Dynamic structures and changing conditions hinder the modeling of such a system with a generic approach.

To keep the overview of every single aspect included during the engineering of a SoS the concept of Model Based Systems Engineering (MBSE) is usually used. It enables stakeholders to gain different viewpoints by abstracting the architecture into different levels. Furthermore, it provides technologies to ensure the availability of an iterative development process. The application of the concepts of MBSE must be assured by a suitable modeling language. Due to its freedom, a so-called General Purpose Language (GPL) can be used in a wide variety of application domains. These language with low constraints is tailored to develop systems working in multiple areas. On the other hand, for describing detailed processes within a certain area, this kind of language is missing specifications. Therefore, the designing of a DSL usually is unavoidable in order to consider all domain-specific features [24]. To increase the effectiveness of the application of MBSE, a well-known approach called Model Driven Architecture (MDA) can be used, which has been introduced by the Object Management Group (OMG). The views specified in MDA are (1) Computation Independent Model (CIM) to provide an understandable description of the system for end users, (2) Platform Independent Model (PIM) to define functionalities and display components of the system, (3) Platform Specific Model (PSM) to formulate interfaces and other technical specifications and (4) Platform Specific Implementation (PSI) to maintain a detailed presentation of code used for describing

components within the system [14].

An example of how to generate interoperability throughout the whole engineering process has already been successfully overcome in the Smart Grid domain. The SGAM has been introduced in order to provide an environment that helps building Smart Grid systems [21]. In [19] the design and implementation of a tool called SGAM Toolbox is described, which ensures the applicability of the theoretical approach. By doing this, the SGAM Toolbox consists of three major parts:

- MDG Technology, which contains the specifications stated in the DSL and provides them for usage
- Model Templates, which support system engineers by providing a fully modelled example and giving information about specific problems
- Reference Data, that contain information about the matrix used in SGAM and make sure to integrate those information into the model

With the help of the SGAM Toolbox, several international projects have already been realized. Through years of use it has established itself as main technology driver to create Smart Grid systems. Adopting these successful concepts to the industrial area can bring the approach of RAMI 4.0 a major step forward.

3 Approach taken for transfer

As already mentioned, the goal behind this approach is to adopt the already established concepts of the SGAM Toolbox for the scope of RAMI 4.0. Although, in the Smart Grid domain, the process of generating energy and providing it to the customers is hierarchically structured. The energy flow passes through multiple zones including several elements, where information is exchanged only with adjacent elements over defined interfaces. This keeps a specific abstraction level, and therefore, allows the modeling of Smart Grid systems to remain structured and understandable, as required from the design principles "divide and conquer" as well as "separation of concerns". However, cross-linking in the Smart Factory is considerable more difficult due to versatile connections and dynamic changes of elements communicating within the network. Adopting the Smart Grid approach to Industry 4.0 including the integration of new domain-specific features and the problem of outcome validation is a big challenge. To deal with this complexity, a new methodology needs to be developed, where all industrial particularities are considered. A similar approach dealing with this issues using the IIRA is introduced in [18]. However, to create a DSL tailored to RAMI 4.0, a dynamic approach needs to be used, where adaptations of the concept may take place anywhere during the engineering process. The concepts introduced by the Agile Design Science Research Methodology (ADSRM) [3] are tailored to this problem. In figure 2, a visual representation of this methodology adapted to the development of the DSL for RAMI 4.0 is given.

In three phases the designing of a dynamic technology, like the creation of the DSL for RAMI 4.0, can take place. In the analysis phase, the domain is elaborated and requirements derived from the Case Study are specified. Afterwards, the development of the Process Model, the DSL and the Toolbox itself takes place during the implementation phase, resulting in an applicable yields model. The last step is the evaluation of the developed technology towards the problem domain. The big advantage of this method is the loose coupling between the single phases, which allow flexible interactions and therefore changes may take place in every phase without influencing the functionality of the whole process.

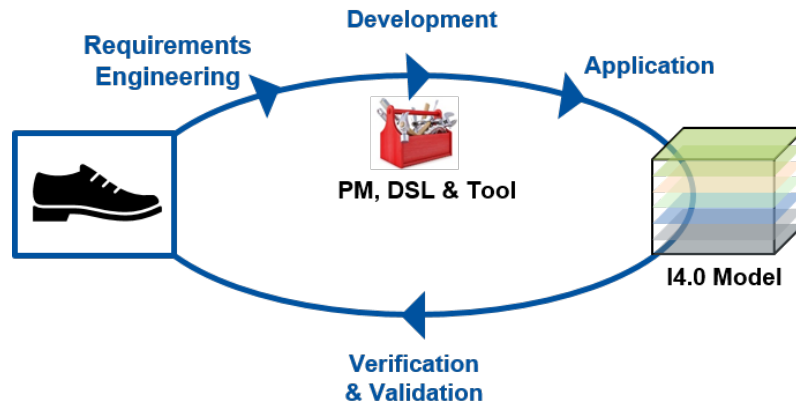


Fig. 2: Agile Design Science Research Methodology for RAMI 4.0

3.1 Case Study Design & 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 shoe manufacturing company that offers the creation of individual shoes to its customers. The manufacturer provides all tools used for customer interaction as well as the factories where the shoes are produced. The goal is to optimize production processes, therefore raw materials and supplier goods need to be available at the time they are used in production. On the other hand, the customer wants to create his individual pair of shoes out of his mind. The ideology of Industry 4.0 is the fully automated processing of the order and the consequent production of the shoes. Therefore, all machines should communicate with each other in order to find the optimal solution concerning resources. Firstly, to keep track of administrative and change efforts, the requirements that the system underlies are elaborated.

Concerning the classification of non-functional requirements, the following five requirements have been specified:

1. **Functionality:** The system to be developed needs to contain all important aspects of Industry 4.0 to allow a detailed and complete description. To achieve this, the framework should support the creator of the system by using well known methods and without raising complexity or administration expenses.
2. **Usability:** Users may come in contact with Industry 4.0 based systems for the first time. Therefore, usage should be clear and supported by demonstration examples as well as automation tools.
3. **Efficiency:** After all, the framework is used to increase productivity, this means that resources should be kept low and time-consuming tasks should be avoided.
4. **Reliability:** The proper creation of a system could be a problem for first-time users. The consideration and prevention of incorrect statements needs to be part of the solution too.
5. **Changeability:** RAMI 4.0 and Industry 4.0 is in consistent change, therefore the framework should be adaptable to these changes. In addition, it should be possible to integrate user-specific solutions in order to react to proprietary implementations.

3.2 Process Model

To manage the creation of industrial models, a specific development process is needed. Technically this process is comparable with the model transformations used by MDA. Furthermore, the single steps of the process are described by the technical processes introduced by the ISO 15288, as depicted in figure 3.

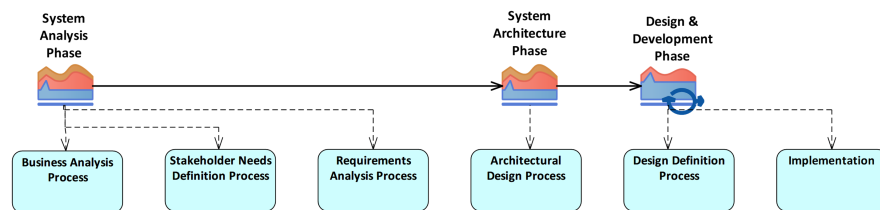


Fig. 3: Development Process for RAMI 4.0 Models

Firstly, the system is analyzed concerning its functionality and business requirements. This functional architecture should give an overview of the system and be understood by people not familiar with the domain but well known with business aspects. Therefore, the aim is to model the basic conditions the system should follow. The system analysis is the base for building a more detailed viewpoint, the

system architecture. It describes the components of the system with their interfaces and connections. The type of connection and technology used to transfer data is the major part designed in this phase of the process. The last step is the detailed modeling of the single components specified by the system architecture. The so-called design of the system displays all elements included and helps dynamically integrating new ones. All in all, this development process deals with an uniform creation of Industry 4.0 models and makes sure that the different abstraction levels of the system are being kept.

3.3 Domain Specific Language

To design a DSL it is important to understand the application domain such as the physical world of Industry 4.0 and CPS. Resulting from this, the behavior and context of the physical domain could be analyzed in order to create a model of the real world. Semantics and structure of this model help defining the abstractions of the Metamodel, dependencies between physical and virtual world formulate the connections of its elements, according to [17]. The Metamodel representing RAMI 4.0 is composed by a conceptual architecture, constituted of the Unified Modeling Language (UML). It describes the conceptional aspects a language needs to contain to model a system based on Industry 4.0. By doing so, the Metamodel is structured in the six layers of RAMI 4.0. On each layer, design elements for describing a viewpoint on a system are provided. The Business Layer therefore consists of elements like business actors, business goals and business cases for representing the cooperation between two actors. With these elements desires of stakeholders can be formulated. High-level use cases are specified to realize business cases on the Function Layer in order to fulfill the defined requirements. Information objects, characterized by a specific data model standard, as well as the connection paths they are exchanged over are being modeled in the lower layers. The Integration Layer offers a representation of the Asset Administration Shell (AAS), a model of the digital twin every physical asset has. Those assets itself are being depicted in the same called Asset Layer.

As the Metamodel is a graphical representation of domain-specific elements and their interconnections, a language is designed for a detailed description of those. Similar to the concepts presented in [4], the conceptual architecture serves as a base to create a specific DSL. This language needs to be utilized throughout the whole development process, from designing the system followed by describing up to modeling it. Consisting of an UML profile, the DSL itself can be designed using well known methods provided by UML. The profile contains all elements previously elaborated from the physical world. Given by UML, the elements itself are consisting of a stereotype and a metaclass. The metaclass is representing the underlying model element where the stereotype is describing the element as it will be used in the DSL.

3.4 Toolbox Implementation

There are several software applications on the market tailored to systems development. Concerning its functionality to extend, the modeling tool Enterprise Architect (EA) developed by Sparx Systems [22] is suitable for providing an environment in order to architect and design Industry 4.0 based systems. To achieve this, the already given general modeling functionalities need to be extended by implementing the DSL. The result is an Add-In called the RAMI Toolbox¹. The main part of this toolbox is the DSL described in the previous section. It consists of the UML profile and two other profiles for the utilization of a tool-set as well as a suitable UML diagram to describe an industrial model. Adapted from the SGAM Toolbox it also provides demonstration examples showing how to use RAMI 4.0. To make use of this DSL, EA needs to load it during its start-up process to provide a set of tools supporting the modeling of industrial systems.

4 Application of the Toolbox

4.1 Case Study Model

The Case Study² itself is created by using the development process described in section 3.2. According to these considerations, the Business Layer contains three major actors, the customer, the manufacturer and the supplier. In the system analysis the goals of each actor are elaborated through requirements engineering. Those goals specify the boundary and rules the system should follow. To keep it simple, this scenario identifies one High Level Use Case (HLUC) "Create Custom Shoes" with the three previously mentioned actors. Out of the generic business model a more specific functional viewpoint can be created. The HLUC is decomposed into more detailed Primary Use Cases (PUCs). "Order Processing" or "Factory Maintenance" could be representatives of this kind. In the Function Layer, every Use Case has actors interacting with them. Figure 4 depicts the most detailed functionalities in the development process like forming supplier goods or assembling raw materials. By doing so, the single functions are represented as Use Cases with their related Actors. The resulting Logical Architecture builds the base for the real architecture of the system including all components and parts. The architectural solution is built referring to the results of the system analysis. The modeled processes need to be represented by physical components. Technological speaking, a model transformation introduced by MDA takes place by mapping Logical Actors to their physical components. In the Information and Communication Layer of RAMI 4.0, the interaction of these components is modelled based on the specifications coming from OPC Unified Architecture (OPC UA). The first step of the system architect is to find

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

² A click-through model is available at <http://www.rami-toolbox.org/UseCaseShoes>

out which kind of information is exchanged between the elements. This process is followed by designing and specifying the communication paths and interfaces of which the information is sent. During this phase the components are seen as Black-boxes and only those needed for interaction are described.

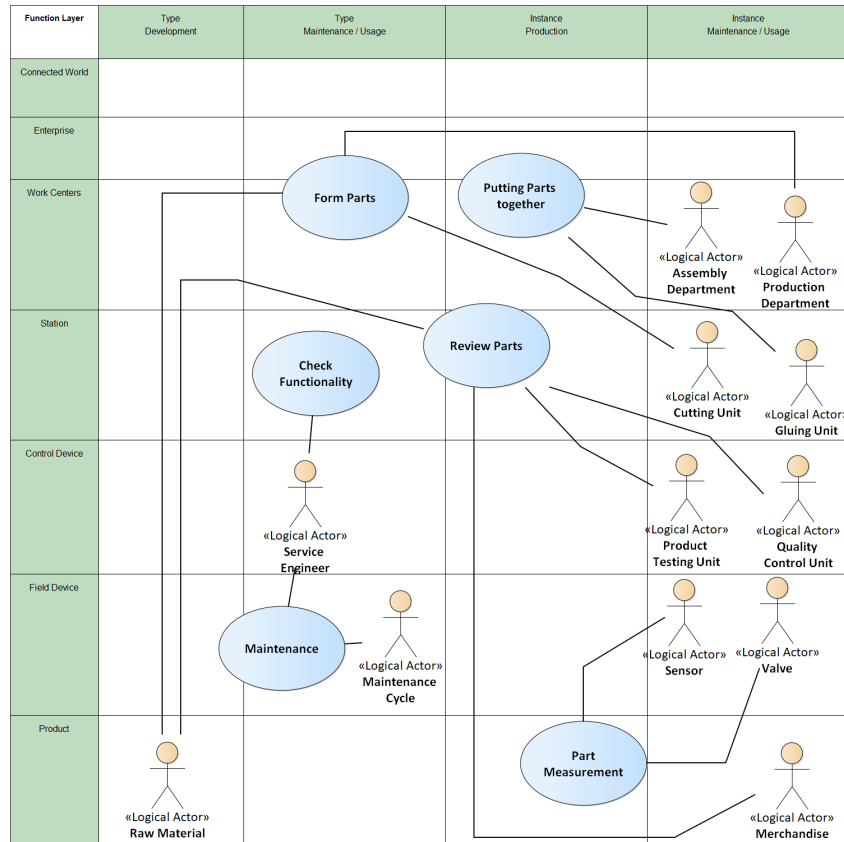


Fig. 4: RAMI Function Layer diagram of the Case Study

The decomposition of the components itself takes place on the Integration and Asset Layer. On these layers, the elements are described as physical units like they are in the real world. The Integration Layer generates a Digital Twin out of the physical units. This means, one AAS containing all information and data as well as safety and security aspects is created for each asset or a group of assets working together. Furthermore, the Integration Layer has to deal with Human Machine Interfaces (HMIs) in order to access the needed data. Technologies like Near Field Communication (NFC), Bluetooth, Barcodes and USB find its place on this layer.

4.2 Findings

Although this generic example enabled the evaluation on a superficial perspective, the used concepts worked fine in general. However, the next iteration step of ADSRM needs to deal with more detailed problems. For example, it was shown that some specifications of DIN 91345 need to be refined or adapted to suit for every domain included. Furthermore, an extension of the process model with familiar standards results in the definition of a more detailed development process. Hence, the standard ISO/IEC 42010 [13] provides a formalization of an architecture framework that may well fit for RAMI 4.0, for example deriving viewpoints and views for each layer. In the same step, the concepts of the Unified Architecture Framework (UAF) standard are elaborated on their suitability for the RAMI Toolbox. The general approach and its enhancements need to be validated by an external domain stakeholder providing a more sophisticated case study in the last step.

5 Conclusions and Future Work

An example of how to deal with modeling and analyzing complex energy systems is the SGAM Toolbox. The already established technology for developing Smart Grid systems has all functionalities needed for system engineering. Due to the similarities between energy and industry domains, the concepts of the SGAM Toolbox [20] may be applicable to Industry 4.0. In this paper, two major concepts have been tested on applicability in the RAMI Toolbox. First, the modeling of Use Cases on basis of an existing reference architecture has been approved by the shoe manufacturing industry example. Although modeling took only place on a superficial perspective, existing concepts and technologies seem to work in the industrial domain as well. The domain specific representation and visualization of components as entry point for discussions or building a common understanding is realized by the DSL. The findings of this paper build a base for the future work of the authors. With the results mentioned above, an application of RAMI 4.0 has been developed for the first time. Now, the results need to be applied to a more sophisticated case study in order to adapt the concept to upcoming domain-specific requirements. In future work, the integration of well-known standards and the development of new features may lead the path towards establishing this approach to become a widely used technology for building Industry 4.0-based architectures.

6 Acknowledgments

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 gratefully acknowledged.

References

1. Bitkom, VDMA, ZVEI: Umsetzungsstrategie Industrie 4.0, Ergebnisbericht der Plattform Industrie 4.0. ZVEI (2015)
2. Boardman, J., Sauser, B.: System of systems-the meaning of. In: System of Systems Engineering, 2006 IEEE/SMC International Conference on, pp. 6–pp. IEEE (2006)
3. Conboy, K., Gleasure, R., Cullina, E.: Agile design science research. In: International Conference on Design Science Research in Information Systems, pp. 168–180. Springer (2015)
4. Dänekas, C., Neureiter, C., Rohjans, S., UsLAR, M., Engel, D.: Towards a model-driven-architecture process for Smart Grid projects. In: Digital enterprise design & management, pp. 47–58. Springer (2014)
5. DIN SPEC: 91345: 2016-04. Reference Architecture Model Industrie 4.0 (2016)
6. Drath, R., Horch, A.: Industrie 4.0: Hit or hype? IEEE Industrial Electronics Magazine **8**(2), 56–58 (2014)
7. Hankel, M., Rexroth, B.: The Reference Architectural Model Industrie 4.0 (RAMI 4.0). ZVEI (2015)
8. Hermann, M., Pentek, T., Otto, B.: Design principles for Industrie 4.0 scenarios. In: System Sciences (HICSS), 49th Hawaii International Conference, pp. 3928–3937. IEEE (2016)
9. Industrial Internet Consortium and Plattform Industrie 4.0: An Industrial Internet Consortium and Plattform Industrie 4.0 Joint Whitepaper (2017)
10. International Electrotechnical Commission: IEC 61512: Batch control (2001)
11. International Electrotechnical Commission: IEC 62264: Enterprise-control system integration (2016)
12. International Electrotechnical Commission: IEC 62890: Life-cycle management for systems and products used in industrial-process measurement, control and automation (2016)
13. International Organization for Standardization: ISO/IEC/IEEE 42010: Systems and software engineering – architecture description (2011)
14. Kempa, M., Mann, Z.A.: Model Driven Architecture. Informatik-Spektrum **28**(4), 298–302 (2005)
15. Lightsey, B.: Systems engineering fundamentals. Tech. rep., DTIC Document (2001)
16. Lin, S.W., Miller, B., Durand, J., Joshi, R., Didier, P., Chigani, A., Torenbeek, R., Duggal, D., Martin, R., Bleakley, G., et al.: Industrial Internet Reference Architecture (IIRA). Industrial Internet Consortium (IIC), Tech. Rep (2015)
17. Mezhyuev, V., Samet, R.: Geometrical meta-metamodel for cyber-physical modelling. In: Cyberworlds (CW), 2013 International Conference on, pp. 89–93. IEEE (2013)
18. Morkevicius, A., Bisikirskiene, L., Bleakley, G.: Using a systems of systems modeling approach for developing Industrial Internet of Things applications. In: System of Systems Engineering Conference (SoSE), 2017 12th, pp. 1–6. IEEE (2017)
19. Neureiter, C.: Introduction to the SGAM Toolbox. Tech. rep., Technical report, Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control, Salzburg University of Applied Sciences (2013)
20. Neureiter, C., Engel, D., Trefke, J., Santodomingo, R., Rohjans, S., UsLAR, M.: Towards consistent Smart Grid architecture tool support: From Use Cases to Visualization. In: Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2014 IEEE PES, pp. 1–6. IEEE (2014)
21. Neureiter, C., UsLAR, M., Engel, D., Lastro, G.: A standards-based approach for domain specific modelling of Smart Grid system architectures. In: System of Systems Engineering Conference (SoSE), 2016 11th, pp. 1–6. IEEE (2016)
22. Sparks, G.: Enterprise Architect user guide (2009)
23. Uhlemann, T.H.J., Lehmann, C., Steinhilper, R.: The digital twin: Realizing the cyber-physical production system for Industry 4.0. Procedia Cirp **61**, 335–340 (2017)
24. Van Deursen, A., Klint, P., Visser, J., et al.: Domain-specific languages: An annotated bibliography. Sigplan Notices **35**(6), 26–36 (2000)
25. Weyrich, M., Ebert, C.: Reference architectures for the Internet of Things. IEEE Software **33**(1), 112–116 (2016)