Towards a Round-trip Engineering Approach utilizing AutomationML for developing complex Architectures of Flexible Production Systems

Christoph Binder
Josef Ressel Center for Dependable System-of-Systems Engineering
Urstein Sued 1, A–5412 Puch/Salzburg, Austria
christoph.binder@fh-salzburg.ac.at

Ambra Calà
Siemens Technology
Otto-Hahn-Ring 6,
81739 Munich, Germany
ambra.cala@siemens.com

Jan Vollmar
Siemens Technology
Guenther-Scharowsky-Str. 1,
91058 Erlangen, Germany
jan.vollmar@siemens.com

Christian Neureiter
Josef Ressel Center for Dependable System-of-Systems Engineering
Urstein Sued 1, A–5412 Puch/Salzburg, Austria
christian.neureiter@fh-salzburg.ac.at

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

***Abstract*—Additive manufacturing and product configurations will become increasingly important in future production systems. Under the term of flexible production systems, single stations should be dynamically switched on or off and development processes are aligned to manufacture products in lot size 1. This trend brings great opportunities for manufacturing companies, but also inherits challenges to overcome. Resulting from this, a heterogeneous tool-landscape has emerged, where each of the single tools is addressing a particular aspect of the value-creation network. An example for such a tool specifically targeting the engineering of such flexible production systems according to the Reference Architecture Model Industrie 4.0 (RAMI 4.0) has been proposed with the RAMI Toolbox. However, as universal frameworks are too general to deal with all aspects of developing such complex systems, other possibilities for conflict-free engineering of the system, like simulation or virtual commissioning, need to be available. Thus, this paper deals with proposing a Round-trip Engineering (RTE) approach of previously modeled flexible production systems according to the peculiarities of RAMI 4.0 and by applying Model-based Systems Engineering (MBSE). Those systems are then processed in other frameworks and further engineered. After being optimized, the results are again reloaded into the RAMI 4.0 system model, where the traceability to the remaining system components is ensured. The chosen methodology for bidirectionally exchanging the engineering data is AutomationML, which allows to store exported information from RAMI 4.0 or import such stored information into it. The RTE-approach is thereby evaluated with a real-world case study, the Siemens Fischertechnik industrial plant model.**

***Index Terms*—Reference Architecture Model Industrie 4.0 (RAMI 4.0), Model-based Systems Engineering (MBSE), Industrial Internet of Things (IIoT), AutomationML, Round-trip Engineering (RTE)**

I. Introduction

Flexible production systems are gaining more and more importance, as they provide manufacturing companies with new business model throughout the whole value creation process. For example, customers could individually configure own products with product configurators, which are subsequently produced in lot size 1. Additionally, workstations or machines like robots could fulfill multiple diverse tasks within the production process [1]. Therefore, they could be conditionally switched on or off and even reconfigured, according to the task they need to take on or whether they are currently needed or not. Those new opportunities are providing a lot of potential to global players or small and medium-sized enterprises (SMEs) to stand out from each other or even remain competitive, but are also accompanied by challenges to overcome [2], [3]. One of these challenges is the increasing complexity within such flexible production systems. In order to automate processes or manufacture products individually, independent decision-making and ubiquitous interconnection is required, which can be reached by integrating Cyber-physical Systems (CPS) or smart products originating from the Industrial Internet of Things (IIoT). However, this leads to an amalgamation of multiple intelligent system components needing to communicate with each other [4] and in further succession a transformation of complicated towards complex production systems [5].

In order to cope with this complexity during the engineering process of such systems, the Reference Architecture Model Industrie 4.0 (RAMI 4.0) has been introduced by several German associations. Originally developed to locate standards within Industry 4.0-based systems, its application area recently expanded and multiple purposes of the three-dimensional cube emerged. One of those applications scenarios is the development of flexible production systems by utilizing the concepts of Model Based Systems Engineering (MBSE) [6]. With this approach, the complexity is counteracted by describing architectures of those systems with system models by applying corresponding framework, the so-called RAMI Toolbox.

However, as this toolbox is especially targeting the architecture development of flexible production systems in terms of basic engineering during the system design phase, it should not be seen as universal framework to be applied for counteracting all challenges throughout the whole value creation process. For example, detailed engineering disciplines like lean engineering [7], runtime simulations of cyber-physical energy systems with Mosaik [8], and virtual commissioning based on factory acceptance tests [9], [10] should be executed with other tools focusing on this particular aspect.

Concluding, while engineering the plant topology of flexible production systems should be done with the RAMI Toolbox, additional engineering steps need to be outsourced to other tools. Thus, the modeled system architecture of the plant has to be prepared for those tools, while externally added engineering information should be imported into RAMI 4.0-based architectures again as well. This allows to ensure the traceability for further engineering steps. However, the best solution for storing this engineering information and transferring it to other tools has been proposed with AutomationML, as it was originally developed just to fulfill this purpose [11]. This means, the interconnection between RAMI 4.0 and AutomationML needs to be investigated, since the respective system model formats are not directly correlated with each other. Additionally, the applicability of the RAMI Toolbox for importing or exporting engineering information needs to be assured. This allows to transmit the information from RAMI 4.0 to all engineering tools within the heterogeneous tool-chain throughout the value-creation process as well as importing the results into RAMI 4.0 again, which contributes to data exchange logistics in engineering networks [12].

In more detail, the main contribution of this paper proposes a Round-trip Engineering (RTE) approach for developing complex architectures of flexible production systems. The architecture itself is developed according to the specifications of RAMI 4.0 and by falling back to the concepts of MBSE, while the resulting engineering information is exchanged to other tools with AutomationML. This paper thereby specifically investigates the possibility to exchange this information between RAMI 4.0 and AutomationML by making use of a bi-directional interface, which is provided by the RAMI Toolbox. Hence, modeled systems are exported to an AutomationML-file or externally processed systems are imported from such a file. The Asset Layer should deal as synchronization point, that keeps all information consistent and enables RTE in any direction. The further optimization of system models either in RAMI 4.0 or AutomationML is out of the scope for this particular paper, only needed aspects to ensure the functionality of RTE are addressed in this context. This functionality is subsequently evaluated with a real-world case study, the Siemens Fischertechnik production plant model.

To address these aspects, the remainder of this work-in-progress paper is structured as follows: In Section II, the background about RAMI 4.0 and AutomationML as well as the related work about RTE is explained in more detail. The scientific approach to investigate and evaluate the contribution is outlined in Section III. The implementation of the RTE approach within the RAMI Toolbox itself is delineated in the next section, whose application is further addressed in Section V. Finally, in Section VI the outcome of the conducted study is referred to and a conclusion is given.

II. Related Work

A. RAMI Toolbox

RAMI 4.0 has been introduced to enable the discussion of current or future industrial systems by providing a common foundation. To enable the location of single aspects of the system, three dimensions have been introduced, each of the dealing with a respective aspect. Thus, the interoperability layers introduce six different viewpoints, such an industrial system might have. Those reach from business perspective, over functions to technical aspects of the system. The horizontal axis deals with the system life-cycle and the value creation process, while the vertical axis inherits the classification of industrial components according to the automation pyramid. RAMI 4.0 is seen as one of the most promising technology drivers when it comes to engineering flexible production systems [13]. However, as it applicability is hindered due to missing specifications or only theoretical described characteristics, additional tools enabling its utilization need to exist.

Therefore the RAMI Toolbox has been introduced as Add-In for the modeling software Enterprise Architect (EA) [6]. This toolbox is developed in C# and provides different functionalities supporting the systems engineering process. First of all, a graphical user interface (GUI) gives access to all implemented functions and uses the structure provided by RAMI 4.0 for navigating through the project. In each of the panes, according to the interoperability layer or automation pyramid level, separate diagrams are available for addressing the respective aspect of the system. Therefore, the RAMI Toolbox also gives access to a Domain Specific Language (DSL) including domain-specific elements and connections. This DSL is implemented as Unified Modeling Language (UML) profile within an EA specific Model-driven Generation (MDG)-file. Additional data is stored in respective XML-files or libraries, which is used by implemented functions. Those functions can be accessed via the Add-In and provide automated model transformation, matrix creation or import/export interfaces.

B. AutomationML

AutomationML is a neutral, open, free and standardized data exchange format based on XML [14]. Originally, it has been developed to bilaterally exchange data between engineering disciplines and in the area of Model-driven Engineering (MDE), as seen in Figure 1. As the authors explain [15], the results of different engineering phases within a sequential engineering process could be stored and transferred to each discipline with AutomationML. For example, the models created during the system design phase with elaborating the plant topology, the mechanical system design or its electrical plans could be stored with AutomationML. The same counts for system models used during the system construction its implementation or operation. Additionally, all test plans and specifications could be added to this single point of truth. It is furthermore stated that AutomationML addresses defects and changes within such a MDE environment accordingly, as those are critical factors originating from various stakeholders. As changes in late project phases are often costly and result in high rework effort, AutomationML also deals with synchronizing this engineering data and deals with change management with efficient data exchange approaches [15].

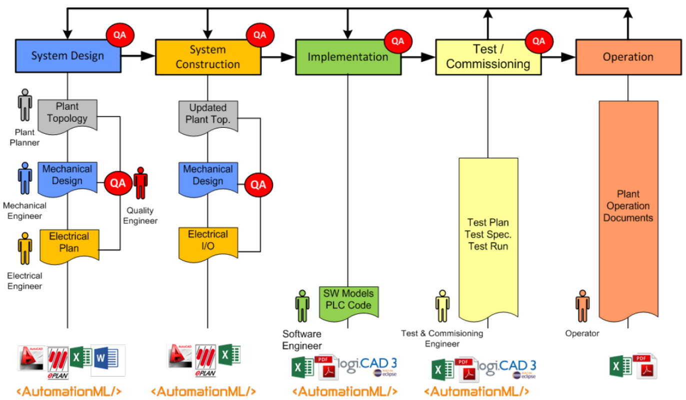

Fig. 1. Sequential Engineering Process with parallel Engineering Activities supported by AutomationML according to [15]

An important advantage can be seen with the object-based arrangement of plant components and their structuring based on CAEX within this standard. This allows to describe objects from higher level perspective and complete manufacturing cells up to single decomposed elements on a lower granularity level. Thereby, the single objects could be derived from abstract classes, while the hierarchical structure enables the definition of sub-elements via composition or aggregation. To do so, AutomationML introduces four major concepts dividing those object-base components. The RoleClasses describe the abstract system architecture and does not consider instances or technical implementation, as those are defined in the InstanceHierarchies. Company-specific RoleClasses could thereby be realized with SystemUnitClasses, while InterfaceClasses define all abstract interfaces or possibilities to exchange data [16]. This XML-based concept enables to associate engineering tools and disciplines in the context of Industry 4.0 or CPS by consistently storing all engineering information within the AutomationML file [17].

C. *Model-based Round-trip Engineering*

The main concept of MBSE is to provide a model to be the main documentation for the system engineering process [18]. Thus, it supports activities throughout the whole system life-cycle, like requirements elaboration or system verification and reduces accompanied risks. Additionally, other activities like simulation and virtual commissioning need to be achieved with model transformations. However, an important aspect of such model-to-model transformations is bidirectionality, in order to keep each model consistent. If changes happen to one of the models, the other model should also adapt accordingly, vice versa. However, this process would require a permanent, bidirectional connection, which is rather complex and therefore often not worth realizing. Thus, a more feasible approach realizing this interconnection is RTE. In RTE, such a connection is not permanently maintained but rather realized through creating the target model from the source model [19]. Thereby, both models need to be consistent, changes to one model should be adapted in the other model as well. Nevertheless, such transformation could not be executed one to one, as either source or target model might include proprietary aspects not being able to be mapped [20]. Thus, the semantics of model changes need to be previously defined and a formal definition for partial or non-injective transformation needs to be available in RTE. On the other hand, this could lead to restrictions in terms of scale.

An example for a successfully implemented process model enabling RTE for adopting and evolving production systems has been proposed in [21]. The introduced “promote-pl” framework offers different process model elements and adaptions for different stages of the RTE process, like product-line management, product-line adoption, product-line evolution as well as domain and application engineering. The main advantage of this process model is to allow practitioners to easily map and even apply RTE activities to production system development processes, as the authors claim.

III. Approach

The approach for developing the RAMI Toolbox has been chosen to be Design Science Research (DSR) as introduced by Hevner et al. [22]. With DSR, a solution for new or unsolved problems could be found in an efficient way, which mostly result in proposing a new theory or creating a novel artifact. Thereby, the toolbox can be considered as design artifact, which is evolutionarily developed. The main advantage of DSR is that such an artifact is not developed at once but in an iterative way by constantly considering new requirements or changes in the environment such as novel fundamental theories. During the research iterations, the toolbox is constantly enhanced and research results are added to the knowledge base. The resulting artifact is then validated against its original purpose or evaluated against the requirements, which is usually done with prototypes or case studies.

While the application of the iteration cycles leading to develop the artifact is not clearly defined, a more applicable methodology needs to be utilized. Thus, the concepts of the Agile Design Science Research Methodology (ADSRM) are taken for use to evolutionarily develop the toolbox. ADSRM itself proposes iteration cycles including five different process steps, where the cycle could be entered in any of them [23]. This allows for this methodology to be applied in various agile application scenarios, like elaborating an RTE approach for RAMI 4.0 by utilizing AutomationML. In this case, the process steps are iterated through in the following way. At first, the case study is specified, which allows to derive requirements for the RTE implementation. This implementation is executed in the next step, followed by the application of the developed artifacts and finally the verification as well as validation of the RTE approach. Concluding, this means, each single of iteration of ADSRM adds an additional aspect to the RAMI Toolbox. While the iteration described in this paper deals with implementing RTE for system models to the toolbox, previous iterations resulted in developing bidirectional interfaces, DSL specifications or architecture definitions. Thus, in the context of DSR the RAMI Toolbox as artifact is increasingly enhanced and enriched with new functionalities.

The first step to enter the iteration cycle is the specification of a case study. To derive requirements and also validate the RTE approach in the context of this paper, the Siemens Fischertechnik industrial plant model is used. This model represents a smart factory that produces plastic housings. To contribute to the topic of flexible production systems, a product configurator is available to individually assemble each plastic housing. The main parts thereby are a base, a cover and an insert, where different variants like a circle or square shaping as well as plugged or screwed tops are available, to mention some examples. To individually produce each plastic housing, the smart factory consists of multiple units like a robot, a gantry crane and flexible assembly lines. The gantry crane could thereby approach four different modules, where various parts of the final product are manufactured. At the bypass, the plastic housings are punched and processed with the help of a robot. While MBSE deals with engineering the plant topology of this Fischertechnik smart factory according to RAMI 4.0, manufacturing process simulations or virtual commissioning of the plant should be done in other tools.

As the main scope of this study is to ensure the applicability of the bi-directional interface between RAMI 4.0 and AutomationML, the described case study is ideal to applied in the context of this paper. The Fischertechnik model deals as proof of concept (PoC) for the conducted study and thus validates the applicability of the RTE method from a superficial perspective. The PoC is thereby representative for any flexible production system and is applied as evaluation strategy in the context of ADSRM. In order to interpret the results of this study, the application of the interface can be considered as successful if the imported/exported system models can be compared and no engineering information is lost during this process. Thus, the main requirement derived from this case study is that the system model of the simulated or commissioned plant should be identical with the system architecture aligned to RAMI 4.0. If not, each other's differences should be recognized and the interface needs to be adjusted to ensure the applicability.

IV. Implementation

A. Case Study Modeling

Before actually implementing the interface between RAMI 4.0 and AutomationML to enable RTE, the Fischertechnik smart factory needs to be modeled according to the specifications of RAMI 4.0. This is done by making use of the MBSE concepts and the already established RAMI Toolbox framework [6]. As delineating the complete model including all RAMI 4.0 layers would exceed the scope of this paper, only the needed aspects to realize the RTE approach are outlined. This means, the Asset Layer, which is describing the physical systems as instances, is used for model-to-model transformations. This layer is resulting from previously modeling the other layers, where requirements, functions or data exchanging are dealt with. Finally, real-world systems are defined, which realize each of the mentioned aspects. As those system components need to be used in other engineering tools, the corresponding model is used for RTE. As seen in Figure 2, the Fischertechnik components are modeled with Systems Modeling Language (SysML) diagrams. In this case, special focus is set on the punching station of the smart factory, which is realized with a SysML block within a SysML block definition diagram.

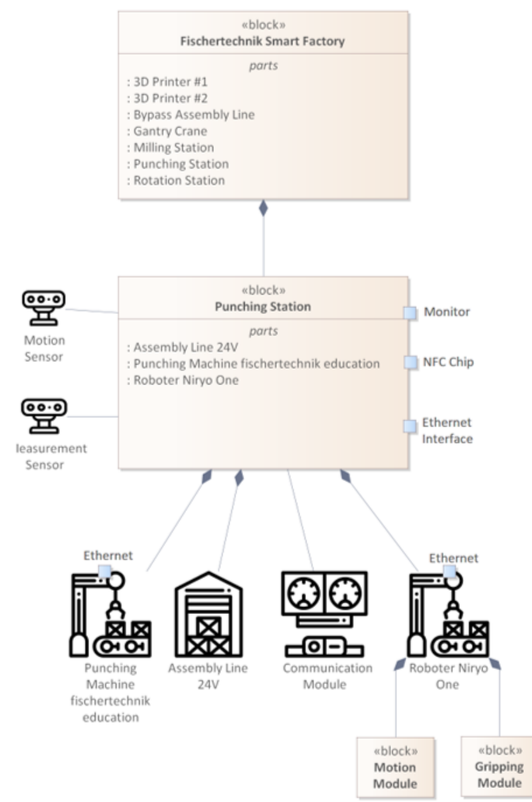

Fig. 2. SysML Block Definition Diagram of the Fischertechnik smart factory

The punching station is part of the complete Fischertechnik smart factory, just as other stations, like a milling or a rotation station. Additionally, two separate 3D printers are included for printing parts of the plastic housing, as well as an assembly line or a gantry crane, which deal with transporting the plastic housing. At a lower granularity, the punching station itself exists of a punching machine, a separate assembly line as well as a robot. In order to interconnect with other system components on the RAMI 4.0 Communication Layer, different communication interfaces, like Near-field Communication

(NFC) or Ethernet, are available, while a measurements sensor or a motion sensor creates events from the punching station within the RAMI 4.0 Integration Layer. Finally, the robot itself contains a motion module as well as a gripping module.

B. RTE Implementation

After developing the model of the case study according to the characteristics of RAMI 4.0, RTE with RAMI 4.0 and AutomationML can be implemented. To do so, an already existing interface between both of the methodologies is expanded with additional functionality, which is subsequently embedded into the RAMI Toolbox. Figure 3 shows the functionality that is provided by the RAMI Toolbox to enable RTE.

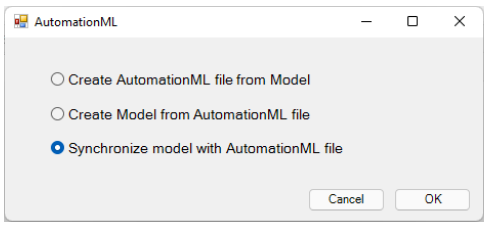

Fig. 3. RAMI Toolbox interface enabling RTE

The interface itself provides three different choices. The two choices on top deal with exporting models to AutomationML files or the other way round. Those are treated in other projects and are not any further explained in the context of this paper. However, the third functionality, synchronizing the model with the AutomationML file, is the one that is needed to enable RTE of RAMI 4.0-based system models describing flexible production systems. This function allows to synchronize independently adapted system models and update them from each other's adjustments. Thus, it provides the foundation for bi-directional model to model transformation in the context of RTE. In detail, when executing this function, the SysML component-tree is recursively iterated through and all components are listed. Subsequently, an AutomationML file is chosen, where also all included system components are discovered. The next step deals with comparing the respective components and find the missing ones in each model. This is done for each part of the component, like interfaces, attributes or sub-components on lower granularity levels. However, if one of the mentioned aspects is missing in the system model or in the AutomationML-file, this functionality creates new model elements and links them to the existing model in each of the tools. Thus, the respective system models are kept consistent with the implemented RTE approach.

V. Application

This section delineates the application of the RTE approach and in the same step the validation of its applicability. By doing so, the corresponding functionality of the RAMI Toolbox[1] is applied to the modeled Fischertechnik industrial plant architecture. Thus, three different scenarios are gone through.

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

The first scenario describes the creation of an AutomationML-file based on the developed system model according to RAMI 4.0. Thereby, the respective SysML diagrams are created within the Asset Layer as a result from MBSE. Subsequently, those models are exported into an AutomationML-file, which is stored to the file system of the operating system. The structure of this file can be viewed in Figure 4. When comparing this image to the SysML model depicted in Figure 2, it can be seen that the developed system is identical to the exported AutomationML InstanceHierarchy model, where it can be used in other tools within the engineering tool-chain, like factory acceptance tests or simulations.

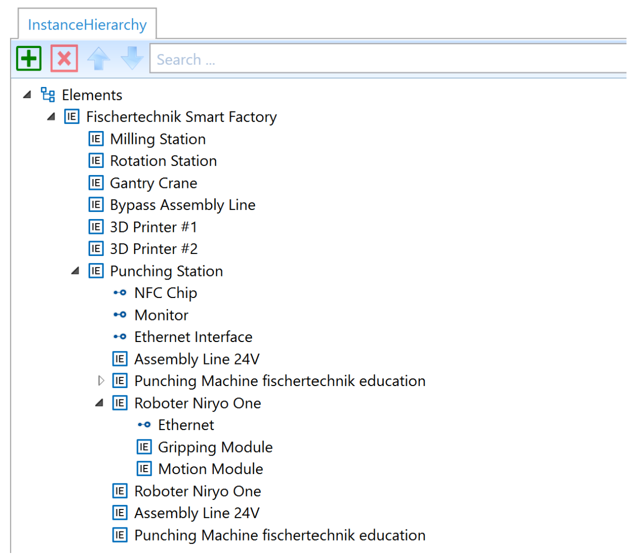

Fig. 4. Fischertechnik industrial plan within AutomationML

The second scenario deals with importing an externally developed system model into RAMI 4.0-layouted architectures. If this model is not yet existing and available in AutomationML structure, this option can be chosen, as it works for each kind of model accessible in AutomationML. Thereby, the SysML block definition diagram is automatically created from the AutomationML-file, where it can further processed within the MBSE development process. Either, further refinements could be done on other granularity levels or the components could be traced to the other layers of RAMI 4.0 to create a comprehensive flexible production system description according to this Service-oriented Architecture (SOA). However, the third scenario summarizes both of the previous scenarios and synchronizes both system models, if they have been edited in each tool, the RAMI Toolbox or AutomationML.

To keep the exemplary case study superficial, no external tools have been used to further process the system. All changes to the system model in AutomationML have been made with the corresponding AutomationML Editor, while the RAMI 4.0-based architecture has been edited with the RAMI Toolbox. All in all, it can be claimed that the described application successfully validates the functionality of the RTE approach utilized within the RAMI Toolbox.

VI. Conclusion & Future Work

Resulting from the increasing complexity of future flexible production systems, different tools & frameworks have been established to process various aspects of those systems, like lean engineering, virtual commissioning or system simulations. This ends in having a heterogeneous tool-landscape throughout the whole engineering life-cycle of such systems. While RAMI 4.0 has established itself to develop the plant topology of such systems with the help of MBSE, other tasks like the previously mentioned ones are not considered in its applicable framework, the RAMI Toolbox. Therefore, this paper describes a RTE approach enabling the usage of RAMI 4.0-specific system models in all other tools included in the engineering tool-chain. As by the term RTE defined, previously modeled system architectures could either be exported and used by other tools or the adjusted system models might be imported into RAMI 4.0 again. Thus, the RAMI Toolbox offers a separate synchronization interface to keep all models consistent, as described in Section IV. The chosen technology for the engineering information transfer to the respective tools is thereby AutomationML, which stores all engineering information within its XML-based structure. The validation towards applicability of the approach is thereby done with the help of a real-world case study in Section V, the Siemens Fischertechnik industrial plant model.

In order to extend the proposed approach towards a ready-to-use methodology for industrial systems engineering, additional work needs to be done. For example, the identification of common concepts in varying engineering disciplines as introduced in [24] and their implementation with the RAMI Toolbox could strongly enhance the proposed approach. Additionally, as soon as SysML 2.0 has been published, the semantics of model interconnections need to be adjusted. Another project on the agenda is the derivation of reference architectures based on RAMI 4.0. This reference architecture implementation and the novel prototype-based version of SysML could deal as base for also considering Role-, SystemUnit- and InterfaceClasses by the proposed RTE approach. This is planned to be done in future iterations of ADSRM and evaluated with more sophisticated case studies.

Acknowledgment

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

References

[1] K. V. Wong and A. Hernandez, "A review of additive manufacturing," International scholarly research notices, vol. 2012, 2012.

[2] L. Safar, J. Sopko, S. Bednar, and R. Poklemba, "Concept of sme business model for industry 4.0 environment," Tem Journal, vol. 7, no. 3, 2018, p. 626.

[3] V. Bittencourt, A. C. Alves, and C. P. Leão, "Industry 4.0 triggered by lean thinking: insights from a systematic literature review," International Journal of Production Research, vol. 59, no. 5, 2021, pp. 1496–1510.

[4] S. Sun, X. Zheng, J. Villalba-Díez, and J. Ordieres-Meré, "Data handling in industry 4.0: Interoperability based on distributed ledger technology," Sensors, vol. 20, no. 11, 2020, p. 3046.

[5] R. Mehr and A. Lüder, "Managing complexity within the engineering of product and production systems," in Security and quality in cyber-physical systems engineering. Springer, 2019, pp. 57–79.

[6] C. Binder, C. Neureiter, and A. Lüder, "Towards a domain-specific approach enabling tool-supported model-based systems engineering of complex industrial internet-of-things applications," Systems, vol. 9, no. 2, 2021.

[7] B. Mrugalska and M. K. Wyrwicka, "Towards lean production in industry 4.0," Procedia engineering, vol. 182, 2017, pp. 466–473.

[8] C. Steinbrink, M. Blank-Babazadeh, A. El-Ama, S. Holly, B. Lüers, M. Nebel-Wenner, R. P. Ramírez Acosta, T. Raub, J. S. Schwarz, S. Stark et al., "Cpes testing with mosaik: Co-simulation planning, execution and analysis," Applied Sciences, vol. 9, no. 5, 2019, p. 923.

[9] M. Al-Saidi and H. Al-Fouzan, "The contribution of remote factory acceptance test towards project's schedule, cost & quality at hawiyah unayzah gas reservoir storage," in International Petroleum Technology Conference. OnePetro, 2022.

[10] M. M. Lopez, A. Bonito-Oliva, P. Valente, T. Boutboul, P. C. de Sousa, A. Loizaga, G. Romano, P. Aprili, R. Harrison, P. Gavouyere-Lasserre et al., "Factory acceptance test and delivery of the first two poloidal field coils to iter fusion facility," IEEE Transactions on Applied Superconductivity, 2022.

[11] S. Faltinski, O. Niggemann, N. Moriz, and A. Mankowski, "Automationml: From data exchange to system planning and simulation," in 2012 IEEE International Conference on Industrial Technology. IEEE, 2012, pp. 378–383.

[12] A. Lüder, J.-L. Pauly, F. Rinker, and S. Biffl, "Data exchange logistics in engineering networks exploiting automated data integration," in 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2019, pp. 657–664.

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

[14] A. Lüder, N. Schmidt, and R. Drath, "Standardized information exchange within production system engineering," in Multi-disciplinary engineering for cyber-physical production systems. Springer, 2017, pp. 235–257.

[15] D. Winkler, F. Ekaputra, and S. Biffl, "Automationml review support in multi-disciplinary engineering environments," in 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, 2016, pp. 1–9.

[16] AutomationML consortium, "Whitepaper automationml part 1 - architecture and general requirements," Tech. Rep., 2014.

[17] L. Berardinelli, S. Biffl, A. Lüder, E. Mätzler, T. Mayerhofer, M. Wimmer, and S. Wolny, "Cross-disciplinary engineering with automationml and sysml," Automatisierungstechnik, vol. 64, no. 4, 2016, pp. 253–269.

[18] A. L. Ramos, J. V. Ferreira, and J. Barceló, "Model-based systems engineering: An emerging approach for modern systems," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 1, 2011, pp. 101–111.

[19] A. Kleppe, Software language engineering: creating domain-specific languages using metamodels. Pearson Education, 2008.

[20] T. Hettel, M. Lawley, and K. Raymond, "Model synchronisation: Definitions for round-trip engineering," in International Conference on Theory and Practice of Model Transformations. Springer, 2008, pp. 31–45.

[21] J. Krüger, W. Mahmood, and T. Berger, "Promote-pl: a round-trip engineering process model for adopting and evolving product lines," in Proceedings of the 24th ACM Conference on Systems and Software Product Line: Volume A-Volume A, 2020, pp. 1–12.

[22] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," MIS quarterly, 2004, pp. 75–105.

[23] 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.

[24] A. Lüder, L. Baumann, A.-K. Behnert, F. Rinker, and S. Biffl, "Paving pathways for digitalization in engineering: common concepts in engineering chains," in 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), vol. 1. IEEE, 2020, pp. 1401–1404.