Automated Model Transformation in modeling Digital Twins of Industrial Internet-of-Things Applications utilizing AutomationML

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*—In recent years, the manufacturing industry sector has undergone major changes. Better known by the term Industry 4.0, this trend describes the transformation of centrally managed production systems towards decentralized value creation networks. As this leads to a strong increase regarding complexity, engineering such current or future manufacturing systems becomes a difficult task. An example for such a challenging engineering process is the selection of technical implementations fulfilling all requirements and performing the needed functions. In order to support this process, model-based systems engineering (MBSE) and its expressions like model-driven architecture (MDA) proved to be promising approaches. However, semi or fully automated model transformations, which could deal with the aforementioned issue, are still yet widely unexplored in the industrial area. Therefore, this paper introduces an approach transforming the logical architecture of a system into its technical implementation by utilizing AutomationML. As the goal is to perform this step automatically, additional tool support is provided by developing specific software. The results of this approach are thereby evaluated by a real-world case study, which is applied according to the concepts of the Reference Architecture Model Industrie 4.0 (RAMI 4.0).**

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

I. Introduction

Designing and developing future manufacturing systems is accompanied by several challenges and difficult tasks. This is primarily attributed to the ongoing integration of mainly intelligent components into traditional production lines. Through continuously introduced results from research and development in the area of Industrial Internet of Things (IIoT) as well as Cyber-physical Systems (CPS), the trend of increasingly automating those systems becomes obvious [1]. However, with the goal to remain competitive over a long term and treat available resources sustainably, different kinds of manufacturing systems are faced with this transformation. One example of this change is the dissolvement of clear structures between the system functions and its technical components [2]. While their interconnection used to be more or less direct within traditional production lines, one function is distributed over many physical elements and one component might execute more than one function within IIoT-based systems. This makes it difficult for system engineers, project managers, machine developers amongst others to find the right technical solution for the previously designed architecture of Industry 4.0 systems [3]. Additionally, performing this process is usually time-consuming manual work [4].

Thus, in order to support system developers in terms of handling the complexity, Model Based Systems Engineering (MBSE) has emerged as a promising methdology to approach these issues [5]. By utilizing models as main information carrier, large-scale, confusing and even complex systems can mutually be developed to consider all stakholder concerns [6]. To ensure the applicability of MBSE and enhance its usability, the Object Management Group (OMG) introduced an extensive approach called Model Driven Architecture (MDA). In order to address the aforementioned problem of finding technical solutions, MDA provides a particular model transformation from the Platform Independent Model (PIM) to the Platform Specific Model (PSM). In [7], it is defined how this transformation is applied on the basis of Reference Architecture Model Industrie 4.0 (RAMI 4.0), as it is considered to be a main technology driver in terms of enabling the transformation towards Industry 4.0. However, while the PIM

978-1-7281-2989-1/21/$31.00 ©2021 IEEE

architecture is precisely specified, the architecture of the PSM lacks specifications.

On the other hand, an established methodology for describing the technical architecture of industrial systems is considered to be AutomationML. With this standard, mutual engineering is driven forward by exchanging plant engineering information across tools and disciplines [8]. As this is desired in industrial MBSE as well, AutomationML is a promising approach to realizing the technical architecture of the PSM. Thus, this paper introduces two major contributions. On the one hand, fully-automated model transformations from the PIM to the PSM, as required by MDA, are developed and provided within the industrial area by utilizing the concepts of RAMI 4.0. On the other hand, the utilization of AutomationML to support such a tool-supported model transformation is validated by applying a real-world case study.

To address these aspects, the remainder of this paper is structured as follows: In Section II, the background about AutomationML and MDA as well as the related work in industrial systems engineering is explained in more detail. The approach to answer the research questions is thereby outlined in Section III. Subsequently, the next section delineates the development and the implementation of the software itself, while its applicability is described and validated in Section V. Finally, in Section VI the results of the conducted study are summarized and a conclusion is given.

II. Related Work

A. Model-driven Architecture

The original goal of MDA is to gain flexibility by having the ability to derive code from a stable model. This is of importance, as the infrastructure within a complex system, like an industrial system, constantly shifts over time [9]. Therefore, MDA proposes different views and equivalent abstraction levels on each view in order to handle the complexity. To change the views' representations, so-called model transformations are introduced [10]. The different views, depicted in Figure 1 are defined as follows [11]: the conceptualization of the system with all requirements, business cases and the system context is modeled within the Computation Independent Model (CIM). A skilled transformation thereby converts those diagrams and its elements to the PIM. There, technical and system requirements as well as constraints of the system are modeled. This means, in the PIM the specification of the system is further defined. Finally, based on automated transformations, those models are transferred into the PSM of the system.

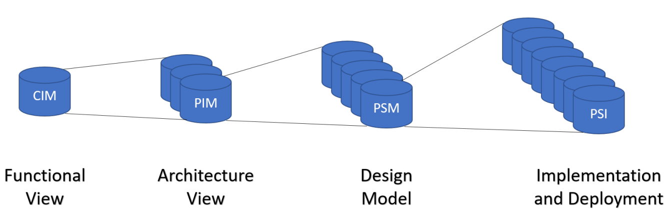

Fig. 1. Model-driven Architecture (MDA)

There, technical specifications as well as the realization of the system is further elaborated. This view also serves as basis for the PSM, that is the code executed on the single components.

B. Industrial Systems Engineering

The approach proposed in this paper is not a completely new topic to talk about. In recent years alone, several publications used MDA for automated model transformations. While some of those approaches focus on automatic data visualization within big data analysis [12], [13], other projects use model transformation based on MDA to transform information within Unified Modeling Language (UML) diagrams [14], [15]. However, it can be observed that recent research focuses on managing the large amount of data within complex infrastructures.

As far as engineering in the manufacturing area is concerned, there have been multiple attempts trying to implement MDA and its concepts. Almost a decade ago, first attempts to handle the increasing complexity in such industrial embedded systems has been investigated based on a radar application [16]. The outcome proves that MBSE is a promising technology driver when it comes to realizing complex distributed and heterogeneous industrial systems. Further research in this area resulted in the introduction of multiple other projects, which were published recently. For example, the application of MDA in wireless sensor networks is proposed in [17], while its utilization for designing logical controllers is further described in [18]. Additionally, another approach specifies the usage of MDA to elaborate knowledge from expertise within industrial safety [19]. The outcome thereby states that models and transformations save a lot of time and other resources within this area.

However, as most of the mentioned approaches focus on single aspects of the system, a methodology for engineering whole industrial systems has been introduced with the RAMI Toolbox [20]. Based on RAMI 4.0, this approach enables mutual modeling and engineering of manufacturing systems according to the concepts of MDA. Nevertheless, as mentioned in Section I, a major shortcoming of this approach as well as other introduced publications is the missing specifications within the PSM. This hinders the automation potential between this model transformation and could be counteracted by the utilization of AutomationML.

C. AutomationML

AutomationML has been developed by several German companies in order to enhance the data exchange between manufacturing engineering tools [8]. Thereby, this standard introduces an object-based arrangement of plant components structured within respective granularity levels. It allows the decomposition into single elements or complete manufacturing cells. To do so, AutomationML makes use of the Computer Aided Engineering Exchange (CAEX) data format, which is based on XML and thus arranges the information accordingly. In summary, the goal of AutomationML is to interconnect engineering tools and disciplines by storing all engineering information following the object-oriented paradigm [21].

Therefore, AutomationML introduces four major concepts of differentiating components within a manufacturing system. The so-called InterfaceClass specifies all interfaces, ports or other data exchange standards within the industrial system. On the other hand, role classes describe the abstract system architecture regardless of its technical implementation. This means, it associates semantics to the respective system element. Next, the SystemUnitClasses need to be defined by users themselves according the available system components. Instance Hierarchies, however, store all project-related information including all utilized instances of system components [22].

III. Approach

Hevner et al. [23] distinguish between two different paradigms while practicing research in the area of information systems. On the one hand, behavioral science aims to predict organizational or human behavior. On the other hand, design science represents the process of extending the boundaries of existing systems by introducing new processes or methods. In this research method, the components to be developed are defined as so-called artifacts. A typical example for such a research artifact within design science in information systems is the approach developed in this work. By utilizing AutomationML for model transformation with the MDA process and trying to fully automatize this, the automation potential and possibilities in such systems significantly increase. The resulting artifact thereby supports future industrial systems engineering in a promising way and could advance applications within the area either of RAMI 4.0 or AutomationML.

Thus, in order to create such research artifacts, a particular framework is proposed, which is part of the Information Systems Design Science Research (IS DSR) [23]. The authors of this method indicate how to develop such a new process or methodology. In more detail, the domain in which the artifact should be located has to be kept in mind by the researcher. Thereby, business models and requirements as well as the surrounding environment need to be known. To create a new artifact, an existing knowledge base and already established methodologies are available for usage in the investigated area. Based on these two inputs the artifact is developed iteratively in the IS DSR. Results of each iteration can thereby contribute to the applicability within the problem domain or be valuable additions to the knowledge base.

As IS DSR itself thus not specify how the iteration cycle for developing the artifacts looks like, Konboy et al. introduced a proprietary research method called Agile Design Science Research Methodology (ADSRM) [24]. By evolutionarily creating the desired process, simple initialization parameters are recommended, which are refined and gain complexity within each iteration. As an exploratory case study is considered to be a suitable entry point to initiate the ADSRM iteration cycle, this paper will use a real-world industrial use case based on a Fischertechnik industrial plant model at Siemens[1].

[1]https://www.fischertechnik.de/

A. Case Study Design

The model of the Fischertechnik simulation thereby consists of different manufacturing processes for creating a plastic housing, like milling or assembling. However, to further enhance the production process, an additional punching machine needs to be implemented. To do so, firstly business cases, requirements and functions are modeled with the help of RAMI 4.0 and the corresponding tool. Next, in order to address the scope of this paper, the engineering process of this implementation is supported by developing a fully automated model transformation from the CIM to the PSM. In more detail, based on the logical architecture, the machines to execute those functions have to be selected. However, as automatically creating the complete technical architecture of the Fischertechnik model would exceed the scope of this paper, special focus on the engineering of engines is set, as this represents a typical industrial scenario. Shortly summarized, the intended approach aims to fulfill the following process. First, during the system engineering process, specific attributes representing the desired performance are added to all functions to be executed by an engine. Based on the chosen values, all available Siemens engines are iterated through and the best fitting ones are selected by an algorithm. Subsequently, a new element for each chosen engine within the PSM of the modeled system is created for usage in further engineering tasks. To fulfill this process, the following types of Siemens engines are chosen:

- Direct current motors: Siemens SIMOTICS DC motors can be installed under difficult conditions due to modular designs.
- Motion control motors: The Siemens SIMOTICS S Servomotors contain several built-in transmitters for optimized motion control.
- High voltage motors: The SIMOTICS HV motors provide a wide range of options, including input voltage as well as rotation speed.

Based on each requirement as well as the functional definition, one of the different categories of motors must be chosen to fulfill the function.

IV. Implementation

This section delineates all implemented steps for enabling automated model transformations. Thereby, three different artifacts have been developed.

A. AutomationML File Generation

The first step towards successfully fulfilling this approach is the need of an AutomationML file. This file contains all available motors and their stereotypes with information about different attributes like voltage, size or rotation speed. Based on this information, the optimal solutions for the previously elaborated requirements are selected. To do so, the AutomationML file consists of different libraries. As defined in [22], standard role classes are required for the modeling of basic AutomationML concepts. Such a class defines the abstract functionality, but does not specify a technical implementation.

Thus, in this paper, all needed information about the motors described in Section III are depicted in the role class library. This also includes the aforementioned attributes, but no concrete values. For each different kind of motor, another role class is created, as different types contain different attributes.

However, in the system unit class library the technical implementations of each abstract role class find their place. This means, the concrete Siemens engines are derived from the respective class and the corresponding values are assigned to the attributes. An overview of the chosen engines and their attributes is illustrated in Figure 2. Those values therefore are the basis for future solution selection from the algorithm. According to the optimal match, the respective engine is suggested to the engineer. As only engines are considered in this case study, no interfaces are taken into account. Thus, there is no interface class library implemented within the AutomationML file. The same counts for the so-called Instance Hierarchy, as this is the representation of the final implemented production system. Nevertheless, in order to provide a full functionality beyond this case study, this file has to be well-kept according to available hardware implementations across manufacturers.

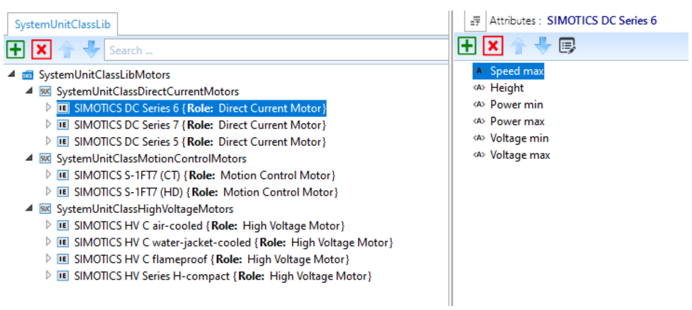

Fig. 2. AutomationML user-specific SystemUnitClass

B. Metamodel Adaption

The current specification of the RAMI Toolbox already inherits a usable meta-model for developing industrial systems according to RAMI 4.0, as described in [7]. However, in order to work with the intended approach, adjustments to the provided modeling elements have to be made. More precisely, especially the Function Layer in the RAMI 4.0 metamodel has to be adapted, which has been done as described in the following. Accordingly, each function representing a drive train must include attributes about desired dimensions, available voltages or the required rotation speed. The concrete values can thereby either be derived from the requirements or calculated from specific mathematical calculations. The results are then embedded into the respective attributes, which have been added to the metamodel representation of RAMI 4.0. A detailed illustration what this looks like in detail and its application is thereby outlined in the next section.

C. Tool Development

The last artifact to enable automated model transformations is the algorithm itself. Implemented within the RAMI Toolbox, the actual functionality needs to follow a specific process. At first, the required attributes are gathered from the logical architecture of the RAMI 4.0 model. Then, according to these values, the AutomationML file is iterated through and the optimized solution is elaborated. Thereby, if there is no optimal solution or more than one solution exists, several trade-offs are considered. Those are presented to the engineer, which ultimately makes a decision. Subsequently, after the right engine has been selected, it is automately created within the technical architecture of the model and traced to the respective function it fulfills. This ensures the traceability within the industrial system and is an important step towards dealing with complexity in such a critical infrastructure [16]. However, when each technical solution has been selected and again implemented into the industrial model, the PSM of the system can be modeled by utilizing the actual applied hardware components. Again, this process is described more precisely in the following section.

V. Application

The Siemens Fischertechnik case study has been chosen to validate the developed artifacts as well as the feasibility of this approach. By doing so, first the metamodel adaption is evaluated by modeling the use case according to the specifications of RAMI 4.0. Subsequently, the model transformation is validated by executing the developed code implementation as well as the AutomationML integration. In the following paragraphs, the process of modeling the case study is described in more detail[2].

By modeling the context as well as the business and manufacturing processes within the Business Layer of RAMI 4.0, a first business analysis is done. Based on this analysis, which can also be considered as developing the CIM of MDA, the goals and requirements for engineering the Fischertechnik model can be derived. In this case, to adapt this model to the concepts of Industry 4.0, several business cases have been derived. For example, the digitization of the manufacturing process, the realization of an Information and Communication Technology (ICT)-network and the integration of the punching machine have been defined. In this example the business case of the machine integration is chosen, as this is suitable for validating the automated model transformation.

In the next step, within the Function Layer, the intended manufacturing process is modeled by introducing high-level use cases. However, multiple granularity levels are utilized by modeling the process with activity diagrams and invocation actions. This enables the considerations of the two modeling paradigms *divide and conquer* as well as *separation of concerns*. The highest granularity level is thereby represented by an activity diagram refining the high-level use case of the intended manufacturing process, as depicted in Figure 3. However, according to the Functional Architecture for Systems (FAS) method, different actions are summarized and traced into Functional Groups, as defined in [25].

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

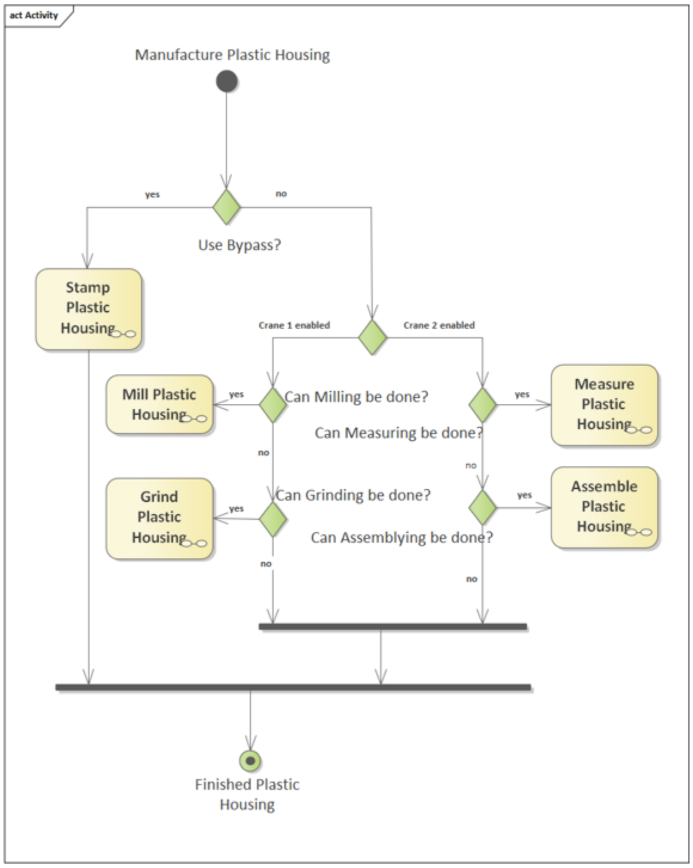

Fig. 3. Intended Manufacutring Process

An example for a summarized element is the bi-directional movement of the plastic housing or turning it in a right angle. However, an important Functional Group in this specific scenario is the drive of the machine. After modeling the corresponding function as black-box and white-box, the values for the attributes of the engine can be calculated. This is where the metamodel adjustment comes into play, which allows storing of those parameters directly within the functional element. In this specific case study, exemplary values of an rotation speed of 4500 rpm, a maximum voltage of 550 kV and a height of 300 mm are chosen. With this step, the development of the PIM is finished.

In the next step, the AutomationML integration and the implemented algorithm within the RAMI Toolbox is used to select technical components to the defined functions. This is done by extracting the information from the attributes and opening the AutomationML file. Based on the gathered information, each available engine within the whole database is then iterated through. In this superficial case study, finding a solution is quite straight-forward as only three values need to be examined. Thereby, the best possible solution is selected and utilized in the technical architecture of the RAMI 4.0. In this specific case, the SIMOTICS DC Series 6 would perfectly fit the previously specified attributes. To ensure the traceability between the PIM and the PSM, the technical component as well as the executed function are connected.

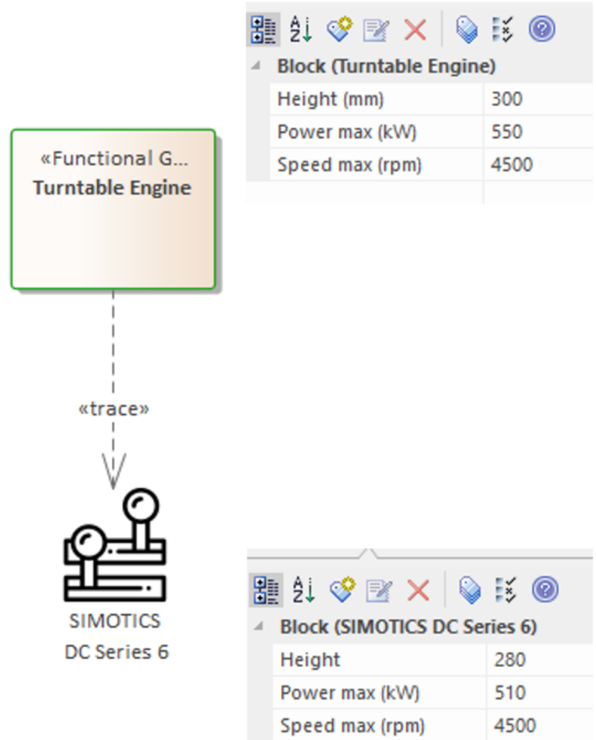

Fig. 4. Sample application run

The outcome after executing one exemplary run with the mentioned settings results in the illustrated overview of elements and tagged values within Figure 4. However, by automatically assigning all chosen technical components to the corresponding modeling elements, the basis for creating the PSM is set. Those automatically created components are used for further modeling the technical specifications of the industrial system. In this case, the bottom four layers provide technical viewpoints, as data exchange is part of the Information Layer, protocols and transmission technologies are part of the Communication Layer and ICT elements as well as Human-machine Interfaces (HMIs) are placed in the Integration Layer. Finally, in the Asset Layer the actual technical specification of each system component is illustrated, which would build the base for a future Instance Hierarchy of the industrial system according to AutomationML.

A. *Findings*

The contribution of this paper indicates, that fully automated model transformations based on AutomationML could be the next step towards industrial applicability of MBSE. Although being validated with a simplified use case, the outlined concepts generally work well in this context. When engineering a system top-down, as the current development process of RAMI 4.0 according to MDA follows, such a method could find broad acceptance within the community.

However, obviously this approach is not a read-to-use methodology, as model transformations based on MDA consist of more complex processes than finding the best fitting component. To validate the feasibility within a whole industrial system, a more sophisticated case study should be applied. Thus, this could be done in the next iteration of ADSRM. Additionally, as AutomationML enhances cooperative engineering across multiple manufacturers, the utilization of capabilities and skills could create better results then the proposed attribute-based approach. Finally, a larger number of attributes could lead to problems by executing the current algorithm. Therefore, methods applied in machine learning or Artificial Intelligence (AI) could deal with more complex environments and constantly improve the results. Nevertheless, this has to be done in follow-up projects and future research plans.

VI. Conclusion & Future Work

As industrial systems become more and more complex, so does the respective systems engineering. By the integration of CPS and IIoT-related aspects, new methods for engineering such systems need to be developed. An example for such an approach is MDA to ensure the applicability of MBSE. Considered to be a future technology driver for engineering Industry 4.0 systems, this method is searching for actual industrial applications. An example for such an application is the selection of optimal technical solutions for previously defined requirements and functions. Obviously, in complex systems, this is a difficult task to fulfill. However, MDA introduces so-called model transformations in order to deal with these issues. Therefore, this paper propses an approach introducing fully automated model transformations from the logical architecture to the technical architecture of industrial systems. The promising standard AutomationML thereby supports this process by providing metadata for each of the machines in extensive databases. The feasibility of this approach is thereby evaluated with the help of an actual industrial case study, the Siemens Fischertechnik model.

On the basis of the outcome of this approach, several follow-up projects could arise. As explained in Section V, a larger amount of attributes or data in general needs to be processed with enhanced methods. Therefore one project could consider the utilization of machine learning to find optimized solution for the elaborated requirements. Additionally, another project needs to investigate the interoperability across manufacturers by utilizing machines or technical components from different production companies. All in all, those projects have to be validated with a more sophisticated case study.

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] D. Kolberg and D. Zühlke, "Lean automation enabled by industry 4.0 technologies," IFAC-PapersOnLine, vol. 48, no. 3, 2015, pp. 1870–1875.

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

[3] T. Wagner, C. Herrmann, and S. Thiede, "Industry 4.0 impacts on lean production systems," Procedia Cirp, vol. 63, 2017, pp. 125–131.

[4] S. Weyer, M. Schmitt, M. Ohmer, and D. Gorecky, "Towards industry 4.0-standardization as the crucial challenge for highly modular, multi-vendor production systems," Ifac-Papersonline, vol. 48, no. 3, 2015, pp. 579–584.

[5] J. A. Estefan et al., "Survey of model-based systems engineering (mbse) methodologies," Incose MBSE Focus Group, vol. 25, no. 8, 2007, pp. 1–12.

[6] L. E. Hart, "Introduction to model-based system engineering (mbse) and sysml," in Delaware Valley INCOSE Chapter Meeting, vol. 30. Ramblewood Country Club Mount Laurel, New Jersey, 2015.

[7] 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.

[8] R. Drath, A. Luder, J. Peschke, and L. Hundt, "Automationml-the glue for seamless automation engineering," in 2008 IEEE International Conference on Emerging Technologies and Factory Automation. IEEE, 2008, pp. 616–623.

[9] R. Soley et al., "Model driven architecture," OMG white paper, vol. 308, no. 308, 2000, p. 5.

[10] A. W. Brown, "Model driven architecture: Principles and practice," Software and Systems Modeling, vol. 3, no. 4, 2004, pp. 314–327.

[11] A. G. Kleppe, J. Warmer, J. B. Warmer, and W. Bast, MDA explained: the model driven architecture: practice and promise. Addison-Wesley Professional, 2003.

[12] M. Golfarelli and S. Rizzi, "A model-driven approach to automate data visualization in big data analytics," Information Visualization, vol. 19, no. 1, 2020, pp. 24–47.

[13] A. Erraissi and A. Belangour, "An approach based on model driven engineering for big data visualization in different visual modes," International Journal of Scientific & Technology Research, 2020.

[14] F. Deeba, S. Kun, M. Shaikh, F. A. Dharejo, S. Hayat, and P. Suwansrikham, "Data transformation of uml diagram by using model driven architecture," in 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA). IEEE, 2018, pp. 300–303.

[15] E. M. MELOUK, Y. Rhazali, and Y. Hadi, "Model transformation approach according model-driven architecture from bpmn to uml up to ifml," in Advancements in Model-Driven Architecture in Software Engineering. IGI Global, 2021, pp. 172–191.

[16] A. Sindico, M. Di Natale, and A. Sangiovanni-Vincentelli, "An industrial system engineering process integrating model driven architecture and model based design," in International Conference on Model Driven Engineering Languages and Systems. Springer, 2012, pp. 810–826.

[17] M. W. Anwar, F. Azam, M. A. Khan, and W. H. Butt, "The applications of model driven architecture (mda) in wireless sensor networks (wsn): Techniques and tools," in Future of Information and Communication Conference. Springer, 2019, pp. 14–27.

[18] G. Łabiak and G. Bazydło, "Model driven architecture approach to logic controller design," in AIP Conference Proceedings, vol. 2040, no. 1. AIP Publishing LLC, 2018, p. 080003.

[19] A. Yurin, A. Berman, O. Nikolaychuk, and N. Dorodnykh, "Knowledge base engineering for industrial safety expertise: A model-driven development approach specialization," in International Conference on Information Technologies. Springer, 2019, pp. 112–124.

[20] 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, 2019, pp. 1–6.

[21] 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.

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

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

[24] 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.

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