



Towards a domain-specific information architecture enabling the investigation and optimization of flexible production systems by utilizing artificial intelligence

Christoph Binder¹ · Christian Neureiter¹ · Arndt Lüder²

Received: 1 March 2022 / Accepted: 10 September 2022
© The Author(s) 2022

Abstract

The industrial domain is undergoing a major transformation, pushed forward by emerging technologies originating from research or industry. The resulting trend, better known by the term Industry 4.0, advances automation within these manufacturing companies by providing ubiquitous interconnection. This enables the integration of technologies mainly used in the Industrial Internet of Things (IIoT), Cyber-physical Systems (CPS) or Big Data with the goal to optimize production processes or facilitate intelligent decision-making. While those processes are progressively supported by methodologies coming from the area of artificial intelligence (AI) like machine learning algorithms, sustainable and consistent storing of production data becomes increasingly important. Concluding, production systems engineering and information engineering are correlating with each other, as the respective results could be used to the advantage of the respective other discipline. In order to address these issues while developing such flexible production systems, the Reference Architecture Model Industrie 4.0 (RAMI 4.0) has been introduced. However, practical applications are lacking as this standard is mainly described in theory, which makes it difficult to actually apply this framework. Thus, the main goal of this paper is to specify a detailed architecture description of the Information Layer to ensure the practical application of RAMI 4.0, which allows stakeholders to utilize model-based Systems Engineering (MBSE) for developing data aspects of industrial systems on the one hand and enable Information Engineering on the other hand. Supported by the concept of the Zachman Framework, the resulting architecture is applied and validated with the help of a real-world case study.

Keywords Software Architecture · Information Engineering · Industrial Internet of Things (IIoT) · Artificial Intelligence (AI) · Model-Based Systems Engineering (MBSE)

1 Introduction

The transformation of traditional production lines towards intelligent manufacturing systems is a major aspect of the novel trend enabled by the Industrial Internet of Things

(IIoT), better known by the term Industry 4.0. With the goal to best optimize production processes, this transformation is supported by continuously integrating new technologies or methods into such systems. Additionally, sustainable resource handling or the automation of manual and repetitive tasks are major achievements in such intelligent manufacturing systems [1]. This automation potential is mainly enabled by the so-called Cyber-physical System (CPS), which are designed to make decentralized decisions based on ubiquitous interconnection as well as constant data exchange [2]. This also leads to the replacement of the original production-oriented fabrication by technology-based services in the manufacturing domain [3]. Consequently, to address the mentioned aspects within current and future production systems, a number of new technologies, like cloud computing, big data or artificial intelligence (AI) are considered to be promising concepts to be utilized within those systems

✉ Christoph Binder
christoph.binder@fh-salzburg.ac.at

Christian Neureiter
christian.neureiter@fh-salzburg.ac.at

Arndt Lüder
arndt.lueder@ovgu.de

¹ Josef Ressel Centre for Dependable System-of-Systems Engineering, Salzburg University of Applied Sciences, Urstein Sued 1, Puch A-5412, Austria

² Otto-v.-Guericke University, Universitätsplatz 2, Magdeburg D-39106, Germany

[4]. The application of these technologies is thereby ensured by additionally integrating a large amount of communicative sensors and actors into those industrial systems. While this leads to an increased complexity in such industrial systems [5] including dynamic changes during run-time or higher degrees of virtualization [6], traditional network components are reaching their limits when trying to fulfill those requirements.

In consequence to the uprising complexity, new methodologies in the area of systems development or data processing are widely discussed. On the one hand, Model-Based Systems Engineering (MBSE) is considered to be the main technology driver when it comes to develop complex industrial systems [7]. With this methodology, traditional document-based approaches, which are reaching their limits when dealing with this complexity, are replaced by model-driven concepts for interrogatively developing product, process and production system in the context of System Engineering. On the other hand, AI continuously produces new methods to deal with the huge amount of data generated by actors as well as sensors and exchanged over dynamic communication structures within such an industrial system during run-time. These methods range over all aspects of Information Engineering, including the generation, distribution, analysis and utilization of data. While already promising on their own, when combining the ideas of Systems Engineering with the ones of Information Engineering, a powerful methodology for governing current and future industrial systems could arise. As MBSE deals with considering the stakeholder concerns and dealing with the system's complexity by applying the paradigms *separation of concerns* as well as *divide and conquer*, Information Engineering could contribute to data administration, the acquisition of higher quality information and process optimization.

However, accompanied by manufacturing system or process optimizations and the utilization of CPS, a new trend of flexible production or evolutionary system development emerges. That is why a lot of efforts in recent research is spent in the area of digital twins, as recognizable in [8]. With this digital representation of the physical system, shortcomings or optimization potential are identified and countermeasures can be analyzed before implementing the system. Within the system model representing the digital twin, changes to the physical system are flexibly realized within short iteration intervals. As far as predictive models and machine learning is concerned, [9] identified the term of the *concept drift* to refer to such changes in the model of a physical system or process, which allows to select the best models with optimized parameters [10]. Consequently, to provide a standardized framework for developing digital twins of rapidly altering industrial systems, the Reference Architecture Model Industrie 4.0 (RAMI 4.0) has been proposed by several German affiliations. With the three-dimensional

layout, such production systems can be structured and different aspects are considered on varying layers. In particular the Information Layer of RAMI 4.0 thus provides a suitable definition for combining MBSE and Information Engineering. Based on this layer, complex industrial information systems can be designed as well as developed and resulting processes can be optimized with the help of AI. Nevertheless, at the current point of view, the Information Layer appears to be a promising concept looking good on paper, but practical applications are still lacking due to a superficial description and missing application scenarios.

Therefore, this specific paper proposes two contributions. At first, a detailed architecture definition for the Information Layer of RAMI 4.0 is proposed enabling the possibility to create digital twins of complex industrial information systems. In detail, this reference architecture thereby integrates the aspects of the ISO 42010 and specifies stakeholders, concerns, viewpoints and model kinds for developing the information aspect of such production systems and allow the possibility to derive predictive models in the context of AI to optimize production processes. Secondly, a possible application of the developed reference architecture is also proposed in this paper. This allows the application of the Information Layer for actual industrial projects according to a specific development process. To underpin this contribution, four different application scenarios are introduced with the goal to evaluate the usability and indicate possible applications.

To address these aspects, the remainder of this paper is structured as follows: In Sect. 2, the background about RAMI 4.0, MBSE and Information Engineering in general is explained in more detail. The approach to develop the research artifacts and the evaluation strategy is thereby outlined in Sect. 3. Subsequently, the next section delineates the development and the implementation of the reference architecture itself, while its possible application is described and validated in Sect. 5. In the next step, the developed artifacts are evaluated towards the specified quality attributes. Finally, in Sect. 7 the results of the conducted study are summarized and a conclusion is given.

2 Related work

2.1 Domain-specific architecture framework

The goal of RAMI 4.0 has been developed by the Platform Industrie 4.0 to enable the discussion about standards, use cases, norms and other relevant aspects related to the industrial area [11]. One of the major goals of the three-dimensional model is to create a common understanding and a mutual basis for current and future production systems. In order to address this intention, RAMI 4.0 extends beyond the entire value creation process aiming to collect

all technical, administrative and commercial data as well as keeping them consistent. Additionally, to also cope with the complexity, this reference architecture proposes the introduction of design patterns for modeling such multi-agent systems [12]. Due to the big influence of its creators on the German industry, the reference architecture encloses multiple sectors within the industrial area and even has been standardized in the norm DIN SPEC 91345 [13].

By integrating multiple axis as well as layers, a system either is developed as whole or single parts of it are considered in more detail, according to the corresponding Industry 4.0 related use case. In more detail, as seen in Fig. 1, the horizontal axis of RAMI 4.0, the so-called Life Cycle & Value Stream, deals with the different states an asset may have during its time of usage by falling back to the criteria introduced in the standard IEC 62890 [14]. In the second axis, the vertical integration within a factory is represented by the “Hierarchy Levels” based on IEC 62264 [15] and IEC 61512 [16], better known under the term *automation pyramid*. Finally, the top-down arrangement of the layers enables the structuring of the system according to the feature of its components across six *Interoperability Layers*.

2.1.1 RAMI 4.0 information layer

In particular the Information Layer of RAMI 4.0 deals with processing the rules resulting from the Function Layer and realizing them by creating information objects. By doing so, also events from the layers below can be classified and prepared for their usage in system functions. In addition, all kind of data accumulated by the system are located within this layer. This allows to collect, summarize and apply them to create new data or exchange them via interfaces. RAMI 4.0 thereby deals as administration tool to assure data integrity [11].

According to [17], the RAMI 4.0 Information Layer should deal with the following issues:

- Offering a run-time environment for event processing.
- The execution of event-related rules.
- The formal description of rules.
- The persistence of the data representing the models.
- Assuring data integrity.
- Consistently integrating of data from different sources.
- The acquisition of new, higher quality data.
- Providing structured data via service interfaces.
- The transformation of events according to data available for the Function Layer.

However, although RAMI 4.0 itself and its Information Layer have been theoretically defined in detail, their utilization appears to be a difficult task, which hinders their application. Thus, it becomes obvious that a more detailed architecture definition needs to become existing, which allows to utilize the RAMI 4.0 Information Layer for actual industrial projects. In order to enable this utilization, the reference architecture has to be expanded by various concepts, like a particular architecture definition including characteristics to ensure the application of AI for developing industrial systems as well as a specific design process. Those concepts are considered as new innovation and are addressed in the remainder of the Paper, especially in Sect. 4.

2.2 Model-based systems engineering

According to [18], MBSE is the formalized application of modeling to support the activities of systems engineering during design and development of the system. These activities include requirements engineering, architecture development, system analyses, verification and validation, among others. Consequently, a system model is developed during the application of MBSE, which represents an abstract representation of the physical system and therefore can be considered as a digital twin. This model however is created during the design phase of the system and is maintained after implementing it. Thus, the system model accompanies the system from its idea to its dissolution [19]. To best possible support this process, such an abstract system model should exhibit the following characteristics [20]:

- The model as a whole can be assembled of various repositories, which have to be consistent in themselves and appear as single model to the user-
- The system model should inherit different viewpoints.
- The system model should be computationally evaluable and provide an abstract syntax.

In contrast to traditional document-based approaches, MBSE creates a central model, which is shared across all stakeholders and prevents inconsistencies by considering

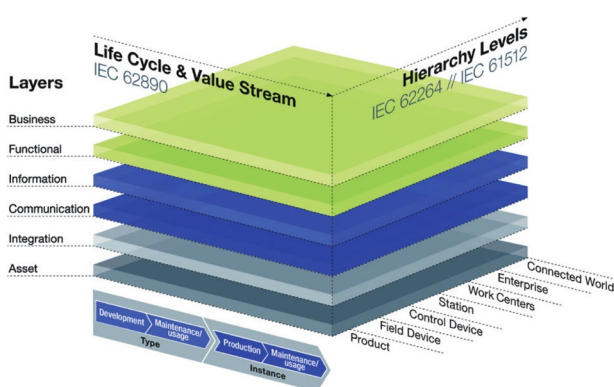


Fig. 1 Reference Architecture Model Industrie 4.0 (RAMI 4.0) [17]

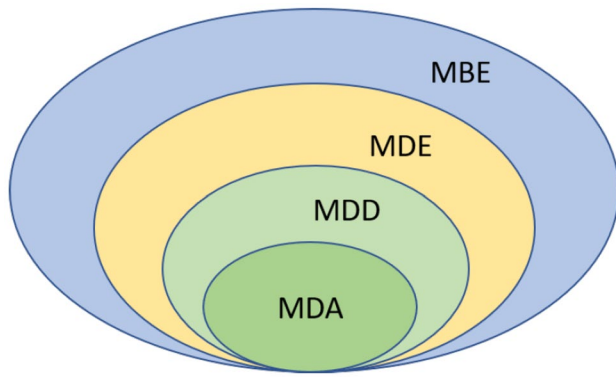


Fig. 2 Dependencies between model-based approaches according to [21]

all relationships between modeling elements when changes occur. To actually apply MBSE in industrial projects, different approaches are provided, as introduced by [21] and visible in Fig. 2. Those are divided into Model-Driven Architecture (MDA), a Object Management Group (OMG)-specific standard how to apply Model-Driven Development (MDD), a general approach to specify models as primary artifacts of systems development. MDD is thereby extended by Model-driven Engineering (MDE) with engineering activities across systems development. Finally, Model-Based Engineering (MBE) refers to the superficial perspective and includes all model-based approaches.

2.3 Information engineering

As defined by Gartner, Information Engineering is the method to develop an integrative information system, which enables the common usage of data to support business decision-making. Thereby, it is assumed that logical data structures and their representation is constant, as it is the opposite with process data, which are used and transferred by processes. Thus, the information system is based on the logical data model while economic decisions are based on process models. According to [22], Information Engineering is defined as:

- The declaration of information itself as a scientific asset.
- The examination of existing methods for information and developing new methods for this purpose. Connections between the methods should be found and analyzed.
- The implementation of available methods within tools and the integration within development environments.
- The application of information engineering methods and tools, followed by an empirical analysis of the respective results.

According to this definition, the concept of Information Engineering is to merge and unify different Applications, which should be elaborated in a complex system or a complex application. This includes technologies like data processing with the help of databases, ensuring controlled data access over interfaces between system components. To achieve this, it is important to use a top-down approach like commonly known in systems engineering, which is evolutionary developed and supported by a framework to sustainably contribute to reaching strategic company goals.

According to [23], four different phases have to be went through to enable Information Engineering in complex systems. Those are aligned according to a pyramid, as seen in Fig. 3. The first step at the top of the pyramid is to do information strategy planning to deal with strategic management goals or define critical factors of success. Then, the processes to execute specific business units are analyzed, which deals as a base for designing the information system, which is the third step of the pyramid. Finally, the single procedures are physically implemented. Consequently, the process of enabling Information Engineering can be compared to the one of systems engineering in general.

After implementing the information system, the single disciplines of Information Engineering can be supported in their execution. Those disciplines reach from data acquisition to data management, data analysis and data representation. In particular in the context of Industry 4.0, the management of data as well as their analysis is of importance. This is substantiated by the fact that the huge number of actors and sensors are producing a large amount of data or business intelligence units need to gain data from different sources. Thus, the keywords of *Big Data* or *Data Warehousing* need to be introduced. As data originating from a data warehouse mostly do not inherit specific value for companies, tools for information analyzing need to be available. Those tools combine data of different quantity and quality to use them for decision-making of strategic, tactical or operative decisions.

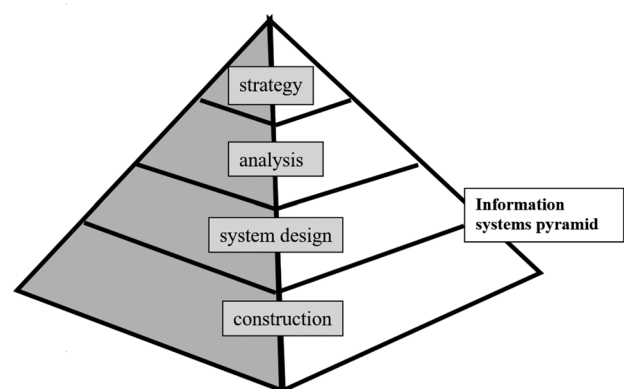


Fig. 3 Pyramid of information systems according to [23]

With the emergence of AI, the task of decision-making is constantly improved to achieve better results. Machine learning techniques or methods from the area of data mining like the CRISP-DM [24] are examples supporting the usage of AI in systems engineering. In conclusion, the Information Layer of RAMI 4.0 needs to consider information management and analysis while designing the architecture industrial production systems.

2.4 State of the art

Industrial Information Engineering is not a completely new topic to talk about. The first steps to drive forward digitalization in production systems has been published by [25]. The work describes modeling a digital twin for CPS according to the RAMI 4.0 administration shell and using the proposed standards. For example, the integration of well-known industrial standards like AutomationML or the Open Platform Communications Unified Architecture (OPCUA) into knowledge graphs is introduced as a successor of the mentioned work. The result thereby indicates that knowledge graphs might be a suitable method when it comes to use data in Industry 4.0 scenarios, to solve the problem of semantic interoperability conflicts.

Additionally, integrating AI into CPS has also been investigated. The authors of [26] propose an architecture, which allows the implementation and usage of AI-algorithms within such systems. While their work gives a first direction how to facilitate the usage of machine learning within CPS, their example is restricted to specific components of the production system, which restricts its usage for a more general application. This is however important in the broad range of manufacturing systems. Another approach also deals with a reference architecture for complex production systems, in this case especially targeting Big Data [27]. This work gives an idea how to deal with all kinds of data in such a system.

Another project is dealing with data analyses in industrial plants with a particular architecture framework [28]. To ensure traceability and quality in system models, the modeling language SysML is used in the context of MBSE. This allows to adapt the model according to new requirements in an easy way. Additionally, the architecture framework itself contains respective solutions for Big Data analyses, data flows or user interfaces. Similar approaches utilizing architecture frameworks based on RAMI 4.0 are applied to find equipment for performing process operations [29] or the integration of security aspects [30].

Continuous and integrated modeling as well as analyses of relevant information within an Industry 4.0 system is the main research aspect of [31]. Their work describes, how dependencies and relationships in such a production system are investigated and illustrated with model-based approaches. The main outcome states that information are

more valuable, if they are available to all participants of the value chain already at an early stage. The authors are also using SysML to model the information system.

A summary of various modeling languages for developing architectures of industrial systems is introduced in [32]. Almost half of the research for the publications is thereby done in Germany. The main topics are thereby dealing with digital representation, system integration and manufacturing processes in general. However, also information handling is an important topic in current and future research. This is substantiated that the number of publications addressing those topics strongly increased within the last decade.

The proposed approaches indicate that MBSE as well as Information Engineering is not a completely new research topic, as it is addressed in various projects. However, as there is a large number of different standards and proprietary solutions, it is difficult to derive a unified approach. Additionally, most of the mentioned projects are solely considering single aspects of industrial systems engineering or data analyses. The key factor to successfully provide a solution contributing to mutual cooperation in this area might be a standardized approach. As RAMI 4.0 represents such a framework providing all needed tools to enable such a cooperation, it is considered as most promising methodology. Thus, in the remainder of this paper, the usage of RAMI 4.0 for such an application scenario is investigated in detail.

3 Approach

As outlined, the goal of this paper is to propose a detailed architecture definition for the Information Layer of RAMI 4.0 to create digital twin architectures of complex industrial information systems and the possibility to optimize the system based on its virtual representation. This includes resource handling, production processes or parameter analyses. As the Information Layer itself provides a suitable foundation for describing the information aspect of such systems, but is not specified in detail in the official documentation whatsoever, a more complete reference architecture description needs to be provided. The research methodology to develop this reference architecture is thereby outlined in the remainder of this section.

In more detail, the utilized research methodology of this paper should consider the rapid rate of change of available technologies or methods in the area of Industry 4.0. Moreover, to apply an agile method is beneficiary for obtaining a possible first result, which is incrementally enhanced in small steps. In addition, it is important to address the problem domain and to make use of already established methodologies. Concluding, the theoretical concepts of Design Science Research (DSR) in information systems appear to be a suitable method for developing applicable results in

the context of this paper. The guidelines introduced by [33] thereby support this development process by addressing new or unsolved problems and identify possibilities to solve them in an efficient way. By applying DSR, the research results are usually innovative and goal-oriented technical solutions, which resolve significant current or future problems. Those solutions are defined as artifacts or theories, as illustrated in Fig. 4.

Those artifacts are developed in an iterative way by constantly adapting to changing requirements or using novel or fundamental theories from the knowledge base. The resulting artifacts should finally be evaluated against the requirements and validated towards usability. This means, the developed solution is examined for possible applications in the respective system environment and researched explored theories or methods are added to the knowledge base. However, the evaluation is typically carried out experimentally using a prototype or a case study.

As DSR itself is a theoretical method and does not provide a specific application process, an agile iterative approach has been developed based on this research methodology. The so-called Agile Design Science Research Methodology (ADSRM) provides just such a method to be applied in the research area of engineering sciences [34]. For usage in different application scenarios, five different process steps are iterated through in one cycle, where the entry point is normally a case study. The resulting design-artifact or theory is evolutionary developed, as recognizable in Fig. 5. In this work, the artifacts are a reference architecture to describe architectures or develop digital twins of industrial information systems in the context of RAMI 4.0, as well as a respective development process, which enables the possible application of this reference architecture. Thus, in the first step, the case study is defined in detail. Based on this case study, requirements for the artifact under development

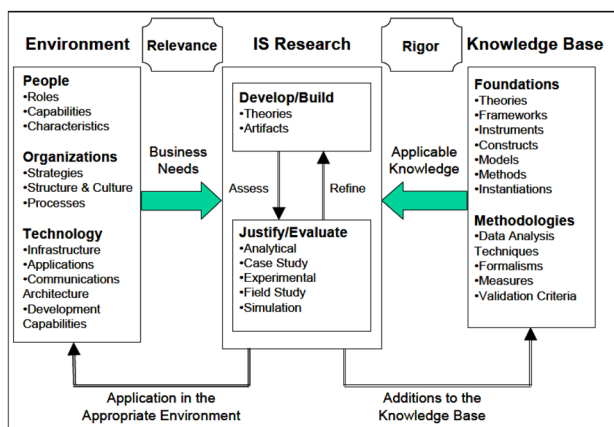


Fig. 4 Design science research in information systems according to [33]

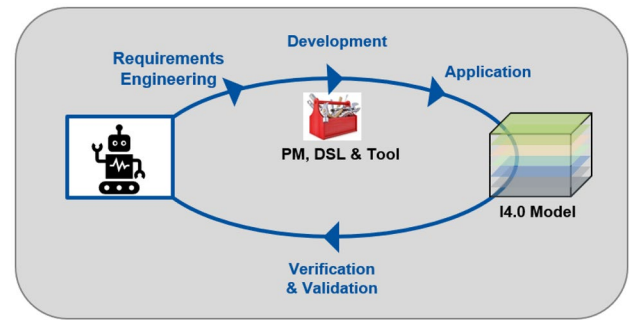


Fig. 5 Agile design science research methodology according to [34]

are derived. Consequently, the artifact is developed and implemented. This means, the reference architecture has to be specified in detail and its application has to be ensured.

Finally, the implemented reference architecture is validated towards feasibility, usability and applicability. Thereby, the concepts of the architecture validation method Software-Architektur Analyse-methode (SAAM) are utilized to validate the resulting reference architecture based on different application scenarios [35]. With this method, the possibility of the described system to fulfill the originally defined quality attributes is investigated. To assure a complete evaluation, outcomes of the developer- as well as the user-context are collected and analyzed. To conclude, the results of each iteration of ADSRM are thereby passed to the next iteration cycle, where an adapted case study or new requirements are building the base for further developments.

3.1 Case study

The research methodology of this paper depends strongly from the case study, as previously explained. Either DSR or ADSRM are using a case study as entry point for research. Thus, the result of this research is dependent from the chosen case study. As the main focus of this work is in the area of Information and Systems Engineering, the use case should be located within this domain. This means, ubiquitous interconnection between the system components, production in sample size 1 and data usage based on AI for designing, producing, utilizing as well as maintaining the production system is encouraged. A major sector of the German industry with a lot of potential concerning the mentioned aspects is automobile manufacturing, which obviously provides a suitable environment for the case study of this paper. Consequently, an interesting concept has been proposed by Audi, where the transformation of original production line manufacturing towards modular production islands is described [36]. The authors explain that future automobile-producing factories consists of up to 200 different production islands, where the car body is maneuvered through. According to

the desired specifications, the transporter only visits those islands, where configurations at the car need to be done. For example, if the customer did not order heated seats, the corresponding production island does not need to be visited. This is expected to lead to an efficiency increase of about 20% for the whole production system [37].

In consideration of the mentioned example, the case study can be described as follows. The powerful competition and constantly changing conditions lead to a high pressure to succeed for most automobile manufacturing companies. Inferring from this, new technologies for process automation or methods to intelligently handle data have to be integrated into value chains. In addition, individual production in sample size 1 or the conversion to hybrid or electric vehicles entail major changes concerning business models or manufacturing processes. Thus, the case study in this work also represents such a transformation of the production network as exemplified by Audi. This includes the model-based development of the modular production islands as well as their interconnection to enable AI based optimizations of parallel production processes. By utilizing this case study it is therefore investigated, how a production system in the automotive industry has to be transformed in order to integrate the concepts of Industry 4.0, especially with regard to the processing of data. In addition, all the systems required to collect, store and analyze information must be designed, modeled and then implemented. This helps the manufacturers to implement the required data infrastructure and to adapt their value creation networks based on the different optimization processes. Moreover, it can be recognized where and which data is generated within the system and how it must be used for sustainable production increases. However, the described case study only deals with the information aspect of the system in the context of this work. The definition of business models, requirements, functional architectures as well as technical aspects would exceed the scope of this paper, which are previously designed and provided by the other layers of RAMI 4.0. The case study itself is depicted in Fig. 6.

As one of the most important aspects of ADSRM is to start the first iteration in a simplified way, a superficial version of the case study is applied in this work. In detail, 10 different production islands are used. These include:

1. Bodywork
2. Paintwork
3. Undercarriage
4. Exterior
5. Transmission
6. Engine
7. Electronics
8. Interior
9. Final Assembly

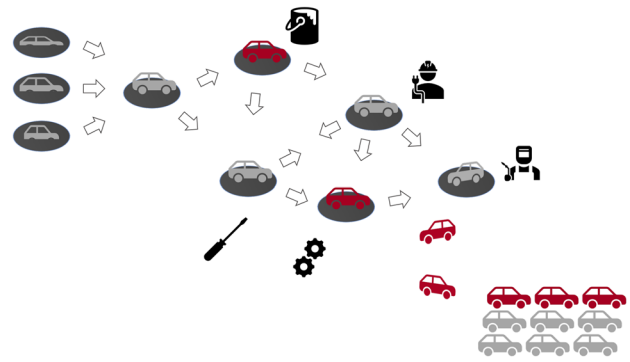


Fig. 6 Case Study according to [36]

10. Quality Check

To also ensure a dynamic development process, dependencies like data exchange or network connections are additionally added. This allows the entire system as well as individual production islands to be examined in detail. The illustration is intended to show which units the case study is made up of and what the modular production line based on it can look like. The production units that manufacture the body can be seen on the top left. Depending on the desired specifications, the car bodies are produced individually. The same applies to the next production unit, where the vehicle is painted. The figure shows that the body can be passed on directly to the painting unit, but also to any other production unit. After the vehicle to be manufactured has been painted, either the undercarriage including the transmission and engine is installed, or the exterior, interior or electronics are installed. In the illustration, two different production units are used for each of the production steps mentioned, each of which can carry out several installations. At the lower right end of the figure you can also see which two production units were used last. After all parts have been installed in the car body, final assembly is responsible for the finishing touches. The vehicle that is ready for delivery is finally checked by a quality control before it is temporarily stored with the other vehicles produced. The control tower in the middle of the graphic assumes the coordination of the transporter with the car body.

For the reasons mentioned, the selected case study allows an investigation of the most varied of scenarios with regard to systems engineering and information engineering. This makes it possible to analyze how the data that accumulates during the production process has to be prepared. In addition, a wide variety of Key Performance Indicator (KPI) can be calculated on the basis of the overall system as well as an individual production island. This also enables to examine the behavior of the system and the interplay of the single production islands to gather

new information. To mention another example, predictive maintenance of single machines is another interesting aspect in the application area of information engineering with regard to the case study.

3.2 Requirements

After defining the case study, the next step of ADSRM and also DSR is to specify requirements for the artifact under development. These business requirements or needs can arise from a wide variety of disciplines and are also relevant for the future area of application. On the one hand, those requirements can be of an organizational nature and can be derived from strategies or processes, on the other hand, people can express their needs to the system in this way. Changed or new technologies can also be drivers for requirements. However, the artifacts to be developed in this work are a reference architecture for the Information Layer of RAMI 4.0 and the possible application of this layer. In this case, a detailed requirement was defined for each of the types of requirement origins described above, which are explained in more detail below:

- **Persons:** The Information Layer should take into account the interests of all involved stakeholders and deal with them in its architecture by introducing viewpoints. To do this, a reference model of this layer should be created, which uses model types and provides specific models aiming to satisfy the respective needs. If possible, standardized and widely accepted methods should be used.
- **Organizations:** The data-oriented company structure and the data exchange between already established processes must be mapped in the Information Layer. It is also necessary to present the individual facts in such a way that all important aspects are included and can be optimized at any time. Ideally, this step should also include the provision of a reference architecture that takes into account the most important aspects of the company with regard to information processing.
- **Technologies:** New technologies should always be used to optimize processes or to adapt economic and structural areas of the company to changing conditions. Especially the trends of Industry 4.0, Big Data and Machine Learning offer new concepts which should be integrated into the Information Layer or can be controlled by it. The Information Layer should therefore offer an interface to up-to-date applications in the field of Information Engineering.

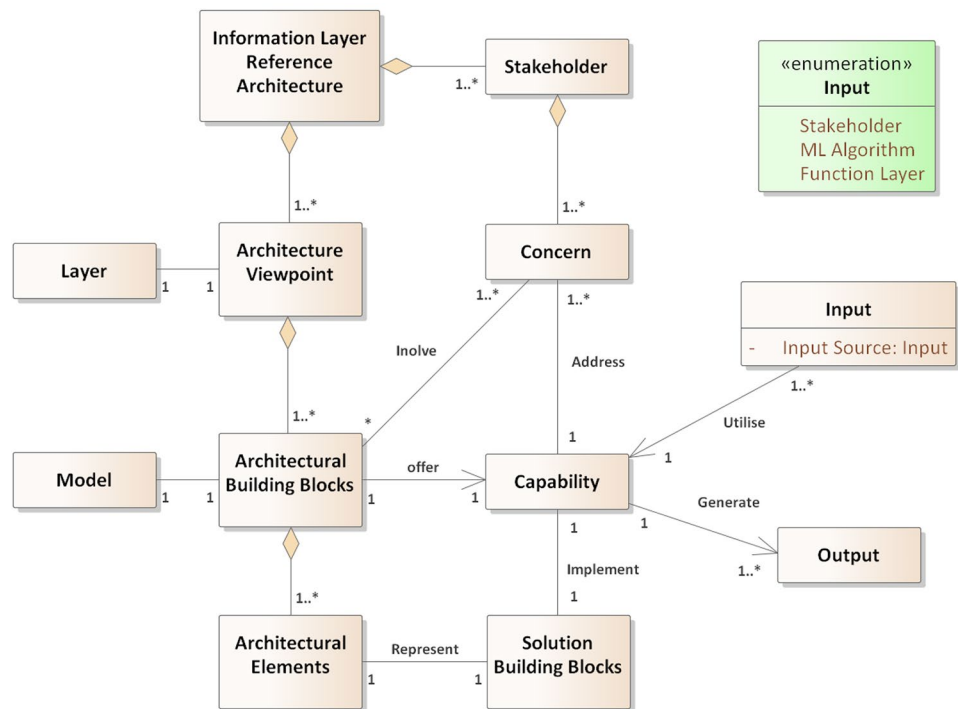
4 Implementation

This section describes in more detail the steps required to implement the RAMI 4.0 Information Layer in practice. Since the Information Layer is hardly defined in theory,

an architecture is first developed which enables the modeling of systems based on this layer. This architecture description should consider the characteristics of RAMI 4.0 and make use of the ISO 42010 basic concepts. In addition, operational decision-making or reporting should be promoted with reference to the modeled data and information. This supports either Systems Engineering or Information Engineering in the context of IIoT. The Information Layer thereby deals as interface between Function and Communication Layer of RAMI 4.0 by defining the information system architecture. This includes the modeling of the information exchange between the individual functional components, the consideration of requirements and standards as well as the implementation of the technical infrastructure. Another aim of the Information Layer is to prepare data for their analysis. Thus, its needs to be expanded by several main characteristics in comparison to its theoretical definition to allow achieving these goals. Those characteristics include a novel metamodel including architectural concepts and modeling elements. Additionally, a particular reference architecture needs to be existing, which provides a blueprint for concrete architectures of actually implemented systems. The next characteristics of the Information Layer architecture are the alignment to established standards like the ISO 42010 as well as the provision of a particular design process. Finally, to enable machine learning analyses, interfaces to external systems need to be provided. As the theoretical definition of the RAMI 4.0 Information Layer does not inherit any of the mentioned aspects, the extension with those characteristics appears to be a major step towards its application and might be considered as main innovation of the paper. Hence, the implementation of the characteristics and the resulting expanded Information Layer ready-to-use architecture are explained in detail in the remainder of this section.

4.1 Metamodel

Before the details of the developed reference architecture for the Information Layer are explained in detail, it makes sense to explain the associated metamodel in detail. As a model of a model, this metamodel defines the concepts and relationships used in the reference architecture. For this reason, it forms the foundation for any instantiated Information Layer architecture. The metamodel developed in this work can be seen in Fig. 7. In the illustrated metamodel it can be seen that the reference architecture of the Information Layer is composed of viewpoints and stakeholders. A single viewpoint is represented by a layer one-to-one, which implements the abstract viewpoint in reality. In this case, a stakeholder can have several concerns, as

Fig. 7 Metamodel of the proposed reference architecture

is also the case in ISO 42010. In addition, a viewpoint consists of several architectural building blocks, which are groups of reusable system elements and can be implemented by using models. These building blocks are chosen in such a way that they take the concerns of the stakeholders into account and include them in the creation of the architecture. The building blocks are also made up of individual architectural elements.

Each of these elements ultimately represents a building block for the concrete architecture solution, the instantiated architectural elements of the modeling language. Another important aspect are the so-called capabilities offered by architectural building blocks. These possibilities also address one or more stakeholder concerns and are implemented using solution building blocks. Every capability utilizes input or generates output, depending on whether the building block captures and uses external aspects and integrates them into the architecture of the system, or whether the architecture generates output from the system. In this work, examples of input into an architecture for the Information Layer are the interests of stakeholders, results from machine learning algorithms or the functional system architecture from the Function Layer.

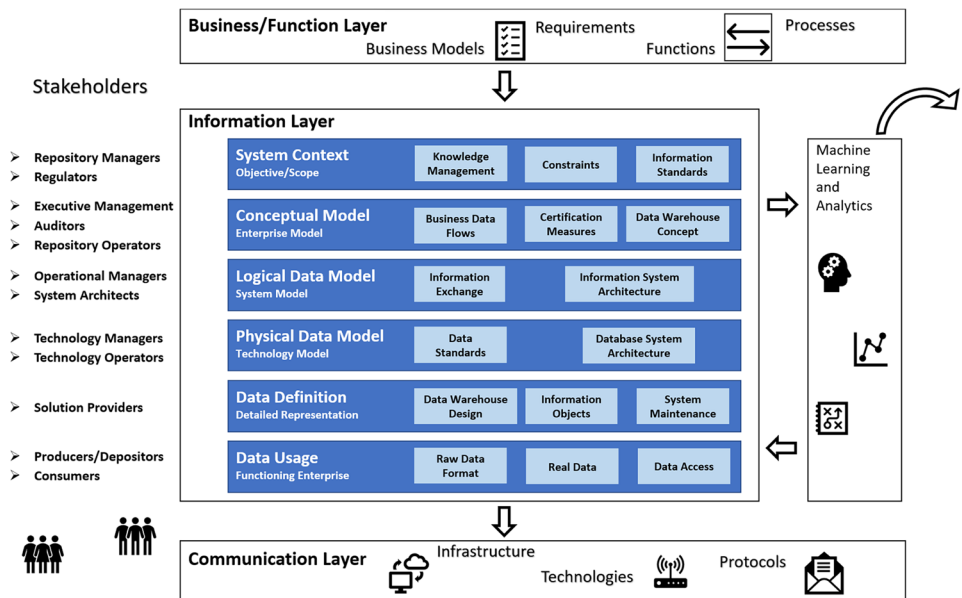
4.2 Reference architecture specification

This section describes the proposed reference architecture for the Information Layer of RAMI 4.0 in detail. As depicted

in Fig. 8, this reference architecture consists of six different layers, which are derived from the Zachman Framework. Those are system context (derived from objective/scope), conceptual model (derived from enterprise model), logical data model (derived from system model), physical data model (derived from technology model), data definition (derived from detailed representation) and data usage (derived from functioning enterprise). Within these layers, the architecture is made up of 16 architectural building blocks, which address various aspects of the information handling of an industrial system. In the context of the model there are different stakeholders who are interested in the reference architecture for the Information Layer. These are aligned to the level of the layer in which the concern exists. However, stakeholders can also deal with certain aspects across all layers, which is not illustrated in the figure. It is also outlined that the reference architecture of the Information Layer must consider system elements such as requirements, business models, functions and processes passed on from above in order to implement the exchange of information. The results are passed on to the Communication Layer in the form of information models, which takes care of ensuring data exchange via standardized interfaces.

Finally, on the right-hand side of the figure, you can see that the data from the system architecture must be made available for a wide variety of methods from the field of machine learning or analytics. The results of these analytical

Fig. 8 Proposed reference architecture of the RAMI 4.0 Information Layer



evaluations can flow back into the model or be passed on to external applications. The core components of the reference architecture are described in detail below. This includes the stakeholders and their concerns as well as the layers and the architectural building blocks of the Information Layer.

4.3 Stakeholder and concerns

According to the ISO 42010, in order to create a successful system architecture, it is necessary to know the stakeholders of the system to be described and to analyze their interests in the system. This enables to create viewpoints and models that satisfy the needs of the stakeholders. In the first step, stakeholders are therefore analyzed from a generic perspective. A wide variety of stakeholders are found who have interests in an industrial system. These are then narrowed down and those who might have potential interests in the Information Layer filtered out. Since the Zachman Framework already contains an established methodology for structuring an information architecture, it is seen as a hopeful approach to the detailed description of the Information Layer. For this reason, the properties of this method are used and combined with those of RAMI 4.0 in order to achieve a complete specification. In addition, the Zachman Framework provides ready-to-use viewpoints and their addressed stakeholders. However, these are also kept superficial and must be adapted to the respective scenario. Therefore, the mapping between RAMI 4.0 and the Zachman Framework can be seen in Table 1.

As shown in the table, the layers of RAMI 4.0 and the Zachman Framework can be mapped directly to one another. The connected world is equated with the system context, the work center with the logical data model and

the field device with the data definition. Only the physical data model of the Zachman Framework can be divided between the two levels of workstation and control device of RAMI 4.0. Additionally, since the Zachman Framework only differentiates between six different stakeholder groups, it is necessary to specify this classification. Therefore, more detailed stakeholders are specified in the next step. The design and development of comprehensive stakeholders of industrial information systems would, however, go beyond the scope of this paper. For this reason, the proposed stakeholders of [38] are used, which imply the roles defined by [39]. The result of this refinement, shown in Table 2, is described in more detail below. The listed stakeholders and their roles are taken up and their concerns in the reference architecture of the Information Layer are more detailed described in this section.

The Stakeholders and their Concerns are described in more detailed in the following. It is important to address these concerns through the reference architecture with architecture modules and to provide solutions for modeling them:

Table 1 RAMI 4.0 Zachman Framework mapping

| RAMI 4.0 layer | Zachman row | Stakeholder |
|-----------------|---------------------|---------------|
| connected world | system context | Planner |
| enterprise | conceptual model | Owner |
| work centers | logical data model | Designer |
| station | physical data model | Builder |
| control device | physical data model | Builder |
| field device | data definition | Subcontractor |
| product | data usage | User |

Table 2 Stakeholders and Roles

| Zachman | Stakeholder | Role |
|------------|---------------------|---------------------|
| Planner | Management | responsible |
| | Regulator | regulator, negative |
| Owner | Repository Manager | decision-maker |
| | Auditor | expert |
| | Repository Operator | operator |
| Designer | Operational Manager | expert |
| | System Architect | consultant |
| Builder | Technology Manager | decision-maker |
| | Technology Operator | operator |
| Contractor | Solution Provider | developer |
| User | Producer/Depositor | operator |
| | Consumer | beneficiary |

- **Management:** This Stakeholder deals with long-term planning the information system. This includes the definition of strategies and goals for the company and its context. It is therefore necessary in this area to interact with all business partners involved or the legal environment. This has to be taken into account especially with regard to data and information of the company. The management therefore has a responsible role.
- **Regulator:** As an external unit, the regulators deal with the corresponding rules for the digital components, such as legal matters or standards. Especially when dealing with information, several aspects must be taken into account, either in relation to the organization or individual technologies. For this reason, the regulatory role has a negative impact on the functionality of the system.
- **Repository Manager:** This Stakeholder plays a responsible and decisive role in making strategic decisions in the organizational area by managing important resources. This leads to the fact that the exchange of information between the business partners or corporate entities is administered and ensured in the long term.
- **Auditor:** The auditors ensure that the company takes into account and implements the rules and standards to be observed. These are mostly experts who contribute their specialist knowledge in dealing with information.
- **Repository Operator:** The repository operator serves as the interface between the business decisions and the detailed design of the system. As a result, he designs the blueprint of the information system to be developed on the basis of the operational role.
- **Operational Manager:** In order to resolve the conflicts between the technical perspective and the business perspective and to make compromises when creating the information system, operations managers can strike a balance between the respective endpoints. These represent the respective interests of the management and

factory employees and can thus act as an interface. This stakeholder therefore has an expert role, which supports technology decisions.

- **System Architect:** The system architect develops the actual architecture of the information system and maintains it. In this way, one or the other decides which components are used where and connects all participating entities.
- **Technology Manager & Operator:** The technology manager decides on the technical means used to ensure system continuity based on advice from the management, while the technology operator maintains and maintains the infrastructure operationally. For example, the manager therefore deals with the different standards and decides on the database system to be selected while the operator introduces it.
- **Solution Provider:** Through the developing role, this stakeholder is concerned with making available all components of the previously defined system architecture. Either these are developed in-house using company resources, or software or platforms are obtained from other providers. Therefore, this role requires the detailed data format, among other things.
- **Producer/Depositor:** The producer deals with the creation of the products intended for the end user. Therefore, this or this must be able to access the different information and its structure in order to be able to use the data for the production.
- **Consumer:** The consumer, on the other hand, uses the end result of the information collected in order to be able to apply it for his or her own interests. This stakeholder therefore needs the respective data.

4.4 Layers and models

After the stakeholders and their concerns have been specified for the Information Layer, the next step in ISO 42010 for architecture development is the definition of viewpoints and models to address the concerns. While viewpoints combine several architectural building blocks and address a larger group of stakeholder concerns, in the simplest case a single building block is created to satisfy a specific concern. This building block is usually implemented using a concrete model. In conclusion, a model can address one or more concerns, but on the other hand it does not necessarily have to address a specific concern and might, for example, only be created as an aid to describing the architecture. Since the Zachman Framework itself already provides elaborated viewpoints, these can be directly applied. The different viewpoints of the system serve the aforementioned stakeholder groups and are arranged accordingly in order to generate different perspectives on the system. From top to bottom, these deal with the important aspects of the company, data models

at different levels of granularity, as well as their definition and physical representation. Therefore, the six stakeholders are used as the basis for Table 3. This table is expanded to include the twelve individual stakeholders and at least one architectural component for fulfilling the concerns is briefly listed in each case.

4.5 Design process

In order to allow a coordinated application of the Information Layer reference architecture, a detailed step-by-step design process needs to be available. Thus, the specification of this architecture also includes a particular development process, guiding users through the modeling tasks. The proposed development process thereby utilizes well-known methodologies, like the established V-model for systems engineering activities. Additionally, as MBSE is utilized to describing such AI-focused production systems, the process steps of MDA are ideal to be applied for step-by-step developing the architectural model of such a system. This means, the design process follows a top-down development strategy and refines systems from a high-level perspective to a detailed technical perspective. Concluding, the design process exists of the following step-by step guideline:

1. Computation Independent Model

(a) Business Data Flow

- (b) Standards & Constraints
- (c) Data Management
- (d) Database Draft

2. Platform Independent Model

- (a) Technical Data Flow & Information Exchange
- (b) Information Systems

3. Platform-Specific Model

- (a) Database System
- (b) Data Standards
- (c) Information Objects & Detailed Data

Within the Computation Independent Model (CIM) the architecture of the information system is defined on a high-level within the problem area, by mainly describing business processes, constraints or desired implementations. Next, as part of the Platform Independent Model (PIM) first possible solutions are given by modeling information flows or abstract systems that handle this information without considering platform-specific details or technologies. Finally, in the Platform-Specific Model (PSM) the technical implementation and final solutions are specified, like databases, used standards or actual exchanged data.

4.6 Development of the external interface

In the end, an interface must be provided so that the required information can be extracted from an instantiated model of the reference architecture of the Information Layer and to prepare the data for the respective external applications. The modeling environment used in this work is the powerful tool Enterprise Architect (EA), which has been developed by SparxSystems. With this software all types of models can be created, preferably in the area of the Unified Modeling Language (UML). EA also provides templates for a wide variety of architecture frameworks or reference architectures in order to promote standardized system modeling. These include the Systems Modeling Language (SysML), Business Process Model and Notation (BPMN) and The Open Group Architecture Framework (TOGAF). Round-trip Engineering (RTE) is also supported by EA. This modeling tool therefore offers a suitable basis for modeling a concrete Information Layer system in order to support selected information engineering methods through the application of the data. The following concepts are therefore used to save data within the model and export it through the interface:

Table 3 Viewpoints and Building Blocks/Models

| Viewpoint | Building Block/Model |
|------------------|------------------------------------|
| System Context | Knowledge Model |
| | Data Flow Model |
| | Standard Model |
| | Constraint Model |
| Conceptual Model | Data Management Model |
| | Certification Model |
| | Database Draft Model |
| Logical Model | Information Exchange Model |
| | Data Flow Model |
| | Inform. System Architecture Model |
| | Dependency Model |
| Physical Model | Data Standard Model |
| | Database System Architecture Model |
| Data Definition | Database System Design Model |
| | Information Object Model |
| | System Maintenance Model |
| Data Usage | Data Format Model |
| | Data Set Model |
| | Data Access Model |

- **Tagged Values:** EA allows the use of the so-called tagged values. This concept allows certain values to be attached to certain model elements. A single tagged value always represents a key/value pair, which stands for identification and value. The tagged values can be created and applied individually for each individual element during the modeling. If these values are already generated in the metamodel, they are consistent for certain types of elements across the entire instantiated model.
- **Attributes:** Class diagrams are a powerful tool in modeling. In this way, they can represent the relationships between the object-oriented system design for the software to be created, but also describe the logical architecture. In the second case, the attributes of the classes store important information about this element.
- **Links:** A model is a powerful communication tool for various stakeholders, but not all information can be stored in it. For example, data sets in the context of big data are too extensive to be stored in the model's database. For this reason, links are used to ensure access to the data.
- **Element Positions:** In order to show the relationships between the modeled elements, their position in the diagram is an important feature. During the modeling process, the elements can be visually linked by creating groups, boundaries or connections.

In order to extract the aforementioned data from the model, several functions are developed in the C# programming language. Depending on the desired functionality, the information is searched for within the model or generated from it. The algorithms themselves are controlled via a user interface, which is illustrated in Fig. 9. The graphic shows the Add-Ins tab in EA, where the expansion of the RAMI 4.0 Toolbox is stored. This add-in provides the additional functionalities. A separate menu item is provided for each of the selected evaluation scenarios for executing them. The process takes place with regular user interaction, in that the current status is displayed or results are presented. The exact application of the functions and their explanation is given in Sect. 6.

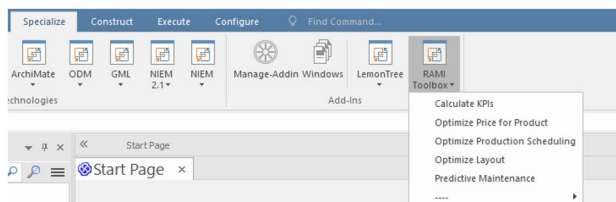


Fig. 9 User interface of the RAMI 4.0 Toolbox

5 Possible utilization

This section describes a possible application of the reference architecture of the Information Layer on the basis of the selected case study. The layers of the reference architecture based on the viewpoints of the Zachman Framework are used and at least one corresponding model type is used for each architectural component in order to address the stakeholders and their concerns. The development of the architecture itself is illustrated from top to bottom using the arrangement of the Zachman viewpoints and the layers of RAMI 4.0. In order to gain closer insights into the methodology of model-based systems engineering in relation to RAMI 4.0 and to recognize the meaning of the Information Layer in this context, possible realizations of the upper layers of RAMI 4.0 are also used as input for the Information Layer as well as models of the Communication Layer are described. This covers both the overall process of industrial system development and the detailed development of the information architecture. While the evaluation of the architecture on the basis of the application scenarios is explained in the next section, this section serves to gain a closer look at the various architecture languages and modeling types. In the individual diagrams, the fulfillment of the interests is discussed in the best possible way and it is explained how the individual solution modules fulfill them. In addition, possible options for recording, managing or analyzing data and information are shown for each of these diagrams. So that the systems engineering process can be treated as completely as possible, an attempt is made to model and map at least one diagram for each layer and each individual architectural building block. The whole model can be found at <https://rami-toolbox.org/InformationLayer/>.

The RAMI 4.0 Business Layer is about describing the requirements for the system to be developed. To do this, the system context, the interaction of the production system with other systems or potential users as well as business processes or business model must first be described. Since the exact modeling of the business architecture is beyond the scope of this work, the result of the business layer is used for the application of the reference architecture of the Information Layer. This result is represented by a requirements model, the elaboration of which is described in detail by [40]. These requirements arise from the business analysis through the system development in the business layer and are passed on to the lower layers of RAMI 4.0. At first, information of high quality must be generated so that a wide variety of evaluations can be carried out in the sense of Industry 4.0. In the second level, it is necessary to collect the information, sustainably store it and then to analyze it. Another requirement of the

production system in this case study is the introduction of modular production units, as required by [37]. In addition, dynamic production processes are required in the future. The process optimization requirement is described on the right-hand side of the diagram. This is generally necessary to automate systems in the direction of Industry 4.0. Therefore, the automation of repetitive processes as well as the ubiquitous connection of the system components is required. The latter can be achieved by implementing the OPCUA technology. Process automation is understood to mean different types of processes. This includes the automation of production processes, administrative processes and engineering processes. When creating a possible architecture of the Information Layer, these requirements must be taken into account.

After the requirements of the system have been defined, the next step deals with developing the functional architecture and defining functions to meet the requirements. Similar to the Business Layer, only the outcome of this layer is used in this work. These results are functions of a production system modeled with SysML and their interaction. The creation of the entire functional architecture follows a specific methodology and the application of Functional Architecture for Systems (FAS) [41]. The functions provide the basis for the exchange of information, since the relationship between the functions is realized via the respective input and output via information. In the Function Layer, this information is described abstractly and from the perspective of the problem area. In the Information Layer, the abstract information is concretized and implemented with regard to the solution area. The Function Layer is therefore the basis for the following engineering activities that are carried out in the Information Layer. In this work and the case study used, the result of the Function Layer consists of a functional model

on three different levels. A separate diagram is created for each level. The diagram of the result of the functional architecture at the highest level of granularity is shown in Fig. 10. This describes the functions at the level of the business level for realizing the main function of the company. Looking at the company in a context diagram including all inputs and outputs as a black box, this diagram serves as a white box to describe all functions within the company in order to convert the input into the output. Since the focus of the selected case study is vehicle manufacturing, all functions are modeled that are required to convert the input into the company's output. To keep this step and other activities simple, only five main functions are used here.

These are the administration of orders, the production planning, the administration of the company, the vehicle production itself as well as the delivery of the vehicles. The required information is exchanged between these functions, such as product specifications, production plans, reports or production data on the vehicles produced. At this level, only the exchange of information is shown, material flows are implemented in production on a lower level. In order to implement external interfaces, the company has a web application function that creates the connection point to customers or suppliers. At the end of this simplified scenario, the vehicles produced are delivered to the respective customer, which is implemented using the function at the bottom right of the graphic.

In Fig. 11, however, the functions on granularity level 2 are described. In this image, the individual main functions of the upper level serve as black-boxes and the process for creating input to output of each of these functions is modeled as a white box. In this scenario, the function of vehicle production is discussed in more detail. Therefore, the information exchange is implemented with the material flow, since

Fig. 10 Functions of the case study at granularity level 1

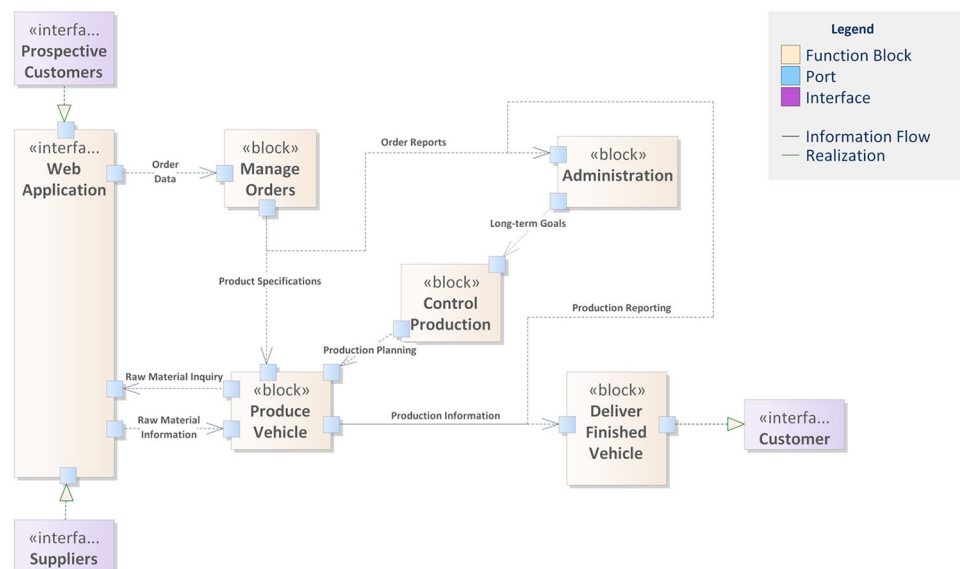
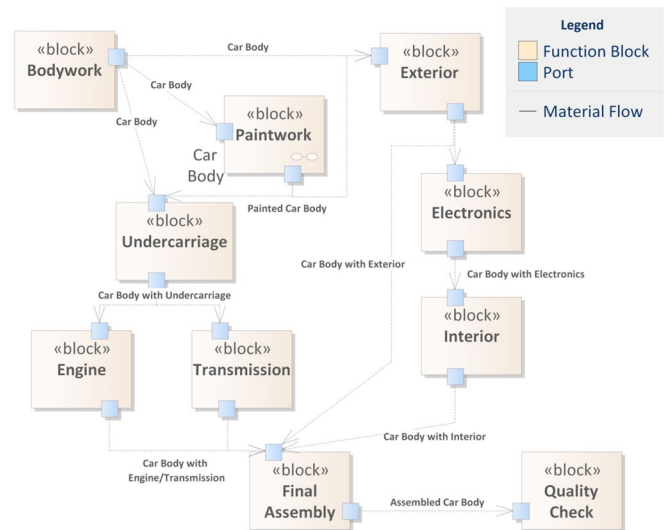


Fig. 11 Functions of the case study at granularity level 2



the transporter of the body contains the information about the specifications itself. The information is requested via this transporter through the respective process step. It follows that this diagram only describes the material flows within vehicle production. As can be seen in Fig. 10, this function requires long-term production planning, the respective product specifications of the vehicles to be produced and information about the available raw materials as input. As a result, the respective information is output as the result of the manufacturing processes with reference to the vehicles produced. AT level 2, the model shows how this input is converted into the respective output.

In the sense of Industry 4.0 and the selected case study, modular manufacturing units are therefore selected for production, which are reduced to ten in this scenario. In the first step, the body of the vehicle to be created is manufactured on the basis of the desired specification. Then, either the paint can be applied, the exterior installed or the undercarriage attached. Depending on the availability or capacity utilization of the production unit and functions that have already been completed, a dynamic decision is made as to which step is to be carried out next. This also applies to the following functions, such as the installation of the transmission and engine, the electronics or the interior. Dynamic production optimizes this compared to assembly line production and thus takes into account the requirements for optimized production processes. The last step in this process is final assembly and the quality check, after which the car is released for delivery.

In order to complete the functions of the functional architecture for this case study, a third level of granularity is introduced. Similar to the second level, a function is selected here and further modeled in detail. In this case, the painting function of the production process is used and described in detail using a SysML diagram, which is shown in Fig. 12.

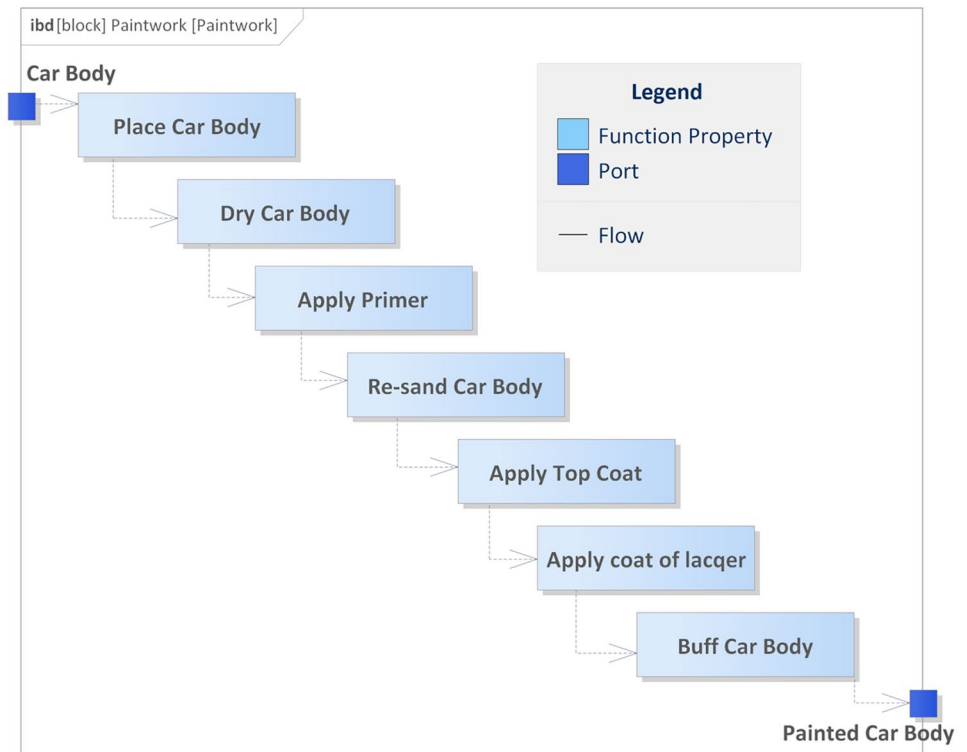
The individual process steps are atomic and will not be further detailed. If necessary, however, these functions can be granulated as required, for which additional granularity levels are introduced downwards. In the context of this case study, three levels are sufficient to evaluate the feasibility of the Information Layer. For this reason, the painting function consists of a linear process with several atomic steps. These are functions such as the placement or drying of the received body, a three-layer painting and a final polishing. At the end, the fully painted body is released for further installations.

The three described diagrams reflect the functions of the case study at different granularity levels. Thereby they serve as input for the Information Layer of RAMI 4.0. With reference to the MBSE, this should define the information exchange between the functions in detail, taking the requirements into account. That is why the architecture of the Information Layer is based on the functional architecture and adds more details to it. While the Function Layer offers a functional view of the production system, the Information Layer deals with the data perspective. In addition, the Information Layer extends the production system to include aspects from the area of Industry 4.0, such as the acquisition of higher quality information, process automation or customized production. With reference to the mentioned subject areas, the reference architecture for the Information Layer should offer a complete concept for its implementation. The extent to which such a concrete information architecture based on the selected case study and taking into account the topics discussed is described in the following section of this section.

5.1 System context

The top layer of the Zachman Framework and thus the Information Layer of RAMI 4.0 deals with the context of the system. More precisely, this includes both the environment

Fig. 12 Functions of the case study at granularity level 3



of the production system of an automobile manufacturer and the context of the information system itself. This means, it is modeled which information arises under which circumstances and how it has to be taken into account. The aim is primarily to find out where the company's important data is located and from which processes important information can arise. From this, it can be deduced how this information must be processed, which happens in the lower layers. Since the stakeholders of this viewpoint mainly act at the business level, the functional diagram of level 1, which contains the essential business functions, is used as the basis for the information analysis in this layer. While technical aspects are preferred on the lower layers, the system context is about the recognition of information and the consideration of rules and standards. Three different building blocks are used for this, which can be modeled as follows based on the case study.

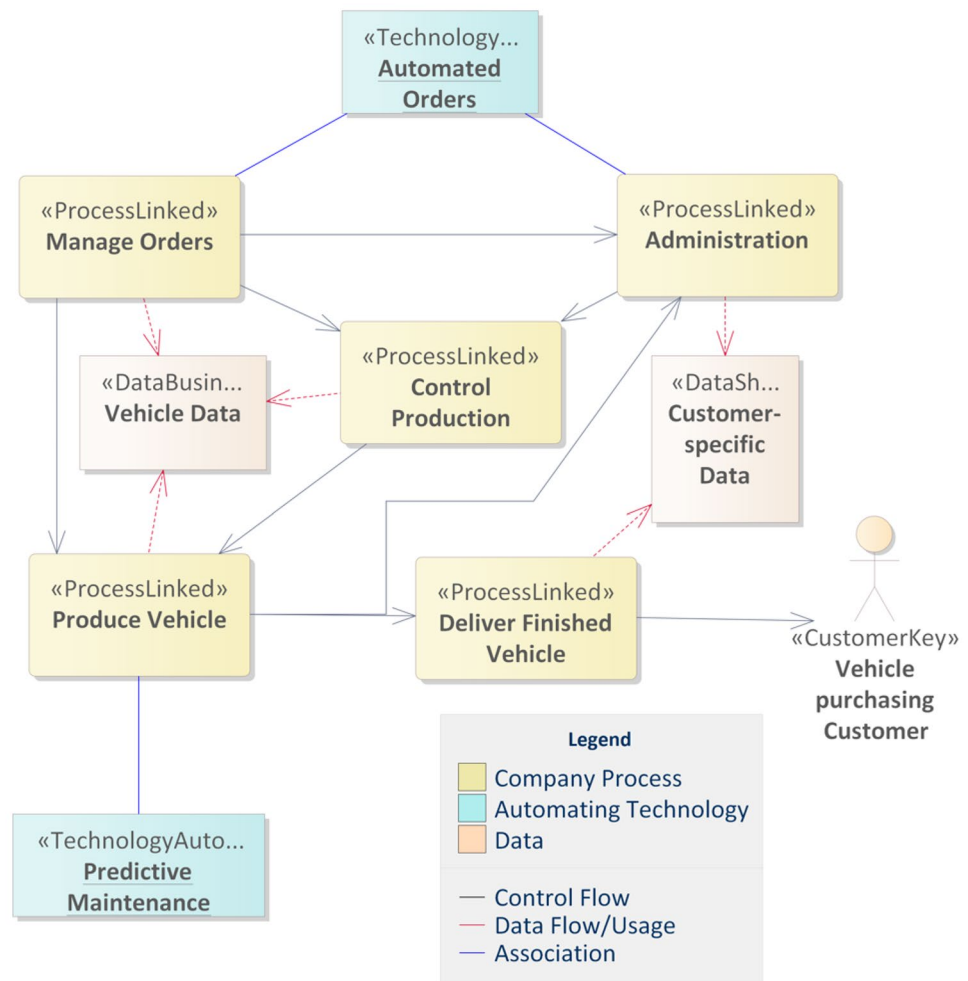
The first diagram at the contextual level describes all the processes, data and technologies that are required for the desired execution of the business process. This applies to ensure the functions defined in Fig. 10. In this way, required data storage or the potential for optimizing the system via information acquisition can be identified. In order to represent these aspects in an understandable way, the so-called Ross & Weill core diagram is used, which enables the representation of the higher level manufacturing process in a diagram, which can be seen in Fig. 13. While the storage of specific data takes place on lower levels of the Information Layer, preprocessing can take place here. For example,

the management can decide which information needs to be stored and how, or how this information is acquired. In addition, this diagram can be used to identify the automation potential of individual processes. This allows responsible stakeholders to understand the required technical architecture of the company and to promote its implementation. In addition, data and processes are recognized at an early stage, which means that the individual system components are prioritized by recording requirements. In the example of the automobile manufacturer, the process from receipt of the order to delivery of the product is described. Based on the functional architecture, the process steps in between are administration, production planning and the production of the vehicles themselves.

The individual processes generate or access customer-specific data and production data. Technical solutions for recording, storing and analyzing this data should be offered below. Another special feature of the Ross & Weill core diagram is the possibility of showing automation potential. For example, within the production process described, machine maintenance can be partially automated through predictive maintenance or raw materials can be ordered independently if there is a scarcity of resources.

In this example, the diagram has been realized by a Domain-Specific Language (DSL), which was integrated into the environment of EA and provides all necessary elements for the description of a complete model for information acquisition. This DSL is a common application mainly

Fig. 13 Potential of information in the production system of the case study



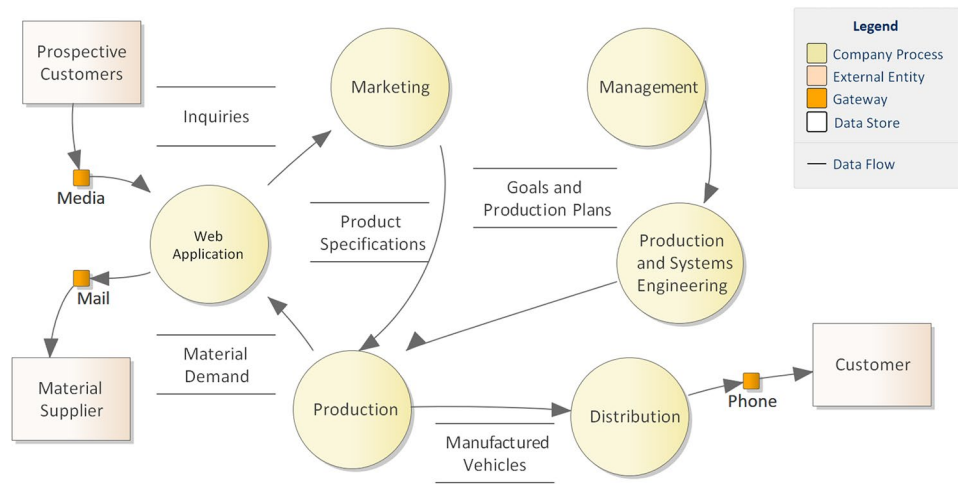
for experts in this domain. For users without access to this DSL the diagram types of the UML are a comparable alternative to the implementation of information acquisition at the highest level. The two concerns of the regulation stakeholder, constraints and standards, also relate to the entire production system at the highest level. For this reason, this diagram is also derived from the corresponding functional architecture.

5.2 Conceptual model

After it has been clarified where the company contains important data and which processes discard which information, the first step is to create a concept for information management. This can be achieved by looking at the exchange of data and the information generated by it within the company. Therefore, the entities of the system between which the data exchange takes place are modeled first. These entities are represented as processes in the data flow model. The individual information that is exchanged between the processes can then be stored in data stores. The purpose

of this model is therefore to recognize which data storage devices are required in the company in order to sustainably store all information arising during the business process. In this case study, the diagram shown in Fig. 14 deals with the data exchange at the highest level, but the data flows in the sub-processes can also be modeled in this way. The picture shown should only give an idea of how a data flow model can be illustrated in detail. Thus, the aim of this model is the representation of the data exchange between the entities and the recognition of the required data storage. At the highest level, this model shows the individual higher level processes of the company that have already been mentioned in the system context and are based on the functional architecture of granularity level 1. In addition, it is specified which data stores are required to ensure the sustainable exchange of data between these processes. Examples of such data stores are the inquiries made, which are passed on from marketing to production, the product specifications which are created by marketing and passed on to production, as well as production plans or digitized guidelines that are created by the management. This diagram also defines how data is exchanged

Fig. 14 Data exchange between the higher level processes



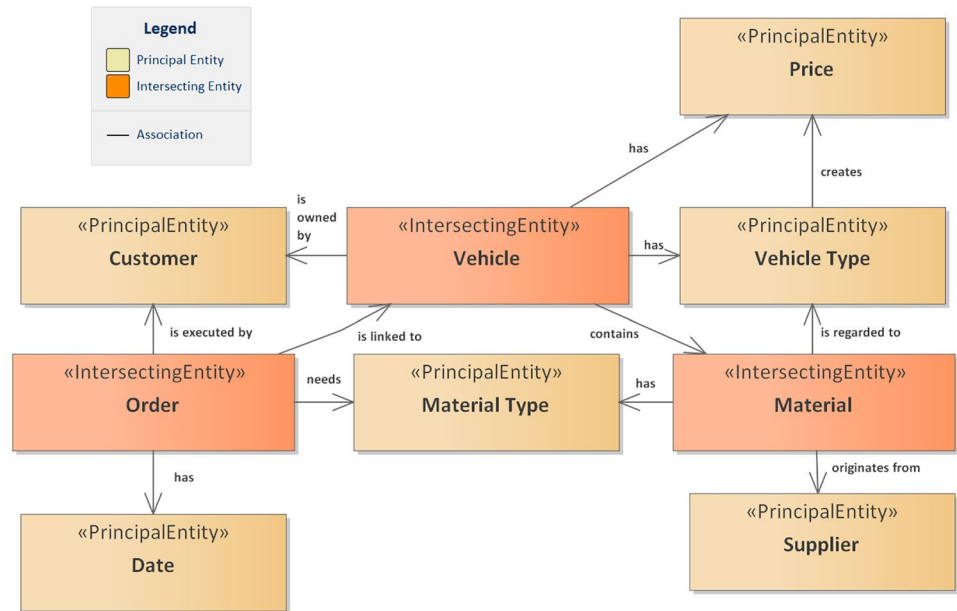
between external partners. The so-called gateways are being introduced for this purpose. In this case study, potential customers are addressed by the marketing department via different media, while production communicates with the suppliers via email. In the specified case study, contact with the end customer usually takes place by telephone. On the basis of the data stores modeled during the data exchange, initial concepts for the data warehouse can be generated. This is done by analyzing the data exchanged at the business level, the potential for information collection and the required data storage and discussing the first options for structuring and managing these data records.

This results in the first common dimensions for the data warehouse, in which it should store and manage the associated data on a sustainable basis. In addition, it is important to find certain similarities from the previously independently modeled diagrams and the data exchange or data analysis in order to ensure data integrity. Therefore, in this step, all recognized possibilities for data handling are listed and the required dimensions are specified, which are referred to as the so-called conformed dimensions. For example, the data store of the inquiries or a single inquiry contains the common dimensions of vehicle type, date and order. A single request also includes the dimension date. The vehicle type, the price and the required material are also used for the vehicle-specific data. The material is also used for the data storage of corporate planning as well as material requirements. The latter needs the material type and the available supplier as well. Finally, the invoices and the customer-specific data are specified. The associated customer and the delivered vehicle type are described here. A date and price are also required for the invoice. As you can see in the figure, there are some redundancies in the dimensions. This is due to the fact that the same data is required for a wide variety of data stores or analyses. In a future data warehouse it is therefore important to take into account the dimensions

identified here. These are summarized vehicle type, date, order, material, material type, price, supplier and customer. For the representation of the selected types a separate DSL is integrated, which supports the data warehouse design according to Kimball.

In order to model a database, the first important step is to define the database schema. Therefore, a conceptual database schema is created before the technical or physical analysis. This database schema shows a section of the real world with all the required properties and relevant relationships. Depending on which section of the real world was selected, the more detailed the associated graphic representation. This representation is usually implemented using an entity-relationship model. The basis for this model are certain specifications or statements in the area of the selected task, which can be specified more precisely through requirements and functions after discussion with the stakeholders. It is not the purpose of this diagram to specify cardinalities for the relationships between the units. This is defined at lower levels in more technical models in order to generate the database from it. In the diagram shown in Fig. 15 it is primarily important to specify the relationships between the entities and to name them individually. This recognizes how the individual dimensions are related and what subsequently has to be taken into account in the data administration. For this reason, the relationships between the previously recognized dimensions are modeled here and thus the relationships between the entities in the data warehouse are shown.

In this case study, the entity-relationship model describes the relationships between the dimensions of the data warehouse and the processes of order collection, vehicle production and delivery of the vehicles, as well as those that overlap several process steps. The most important overlapping entities for this process are the order, the vehicle itself and the material. The main goal of the entity-relationship model is therefore to create an initial concept for structuring

Fig. 15 Entity-relationship-Diagram of the case study

data and assigning it to business processes for later processing. In addition to the respective entities, the relationships to each other are therefore also described in order to later specify primary and foreign keys. The order is carried out by a customer and has a date. In addition, certain material is required for the development of the vehicle associated with the order. The vehicle itself has a price, has a type and is made of different materials. This material comes from a supplier and is related to a specific vehicle type. At this level of granularity, however, it is not yet specified which multiplicity prevails between the individual entities, only the relationship is shown. This model thus serves as an interface between context-relevant objects in the real world and the technical implementation of databases.

5.3 Logical data model

In the logical data model, the system is viewed from a lower granularity level. While the system is seen at the highest point of view in the two upper levels, the actual production system is in the foreground here. In detail, this means that core processes of the company, its departments and the contact with external partners drive the conception of the system. This concept is worked out and described in the logical perspective. Therefore, the functional architecture of the production system on granularity level 2 is used here and used for subsequent analyses in this layer. This defines the exchange of information in the current production system and shows the production processes and the components involved. The specified aspects from the context and the concept are taken into account and a design template is given for the physical system development.

The first step in developing this perspective is defining the manufacturing process. In this way, it can be decided which processes are related and how and by which machines are carried out. The manufacturing process selected in this case study can be seen in Fig. 16 using an activity diagram. The data flows of the individual production units are shown here, which demonstrates the data exchange by implementing control flows. The first two sub-processes deal with the creation of the body and its painting. After this has been completed, either the undercarriage or the exterior can be attached. The material for the undercarriage is passed on via the data flow, while the specifications chosen by the customer for the exterior are passed on to this process. If the undercarriage is selected first, the selected engine and the associated transmission can then be installed, the order is not decisive. If, on the other hand, the exterior is attached, all the electronic components selected by the customer as well as the specified interior are then attached. At the end of the manufacturing process, the vehicle is finally assembled and checked for quality. It is important, which method is chosen for the assembly and which aspects are particularly considered during the quality check.

On the basis of the process model, the dependencies between the individual production units can then be modeled, as shown in Fig. 17.

For the development of the information system architecture, this model shows which units are dependent on each other, which has two advantages. On the one hand, specific restrictions can be set for future optimization algorithms, on the other hand, the sequence for the exchange of information can be set. That is why this model addresses several stakeholders. In this case study, the dependency model

Fig. 16 Manufacturing process of vehicle manufacturing

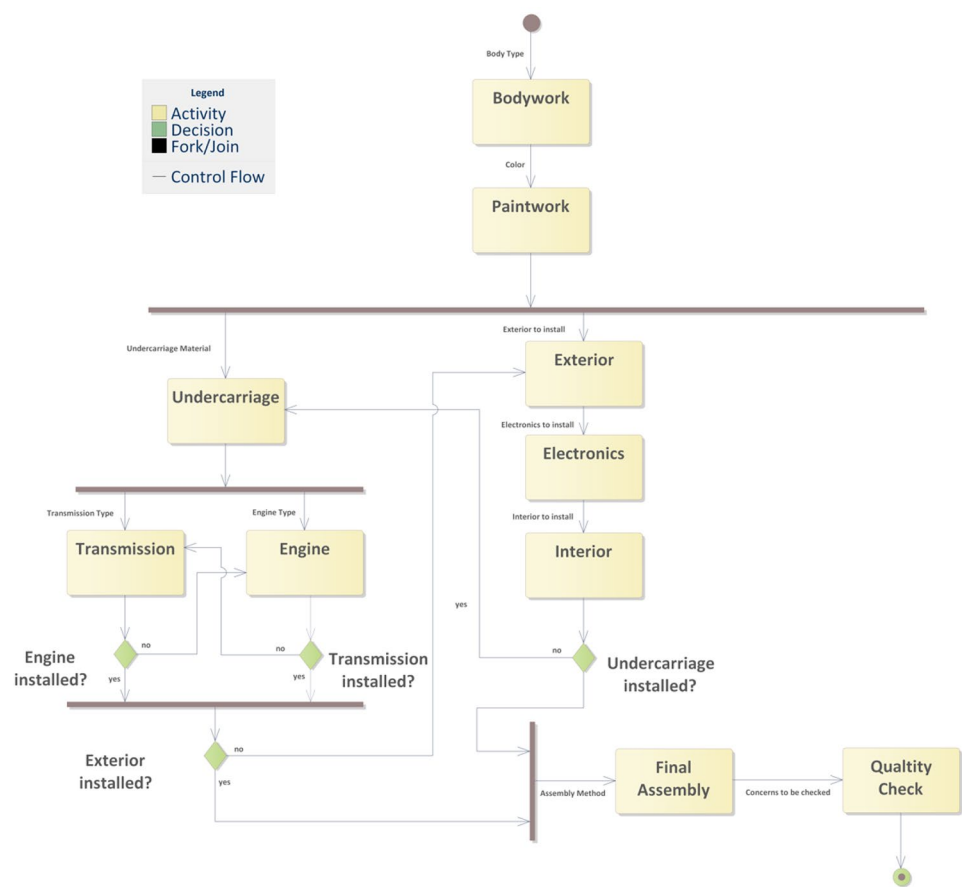
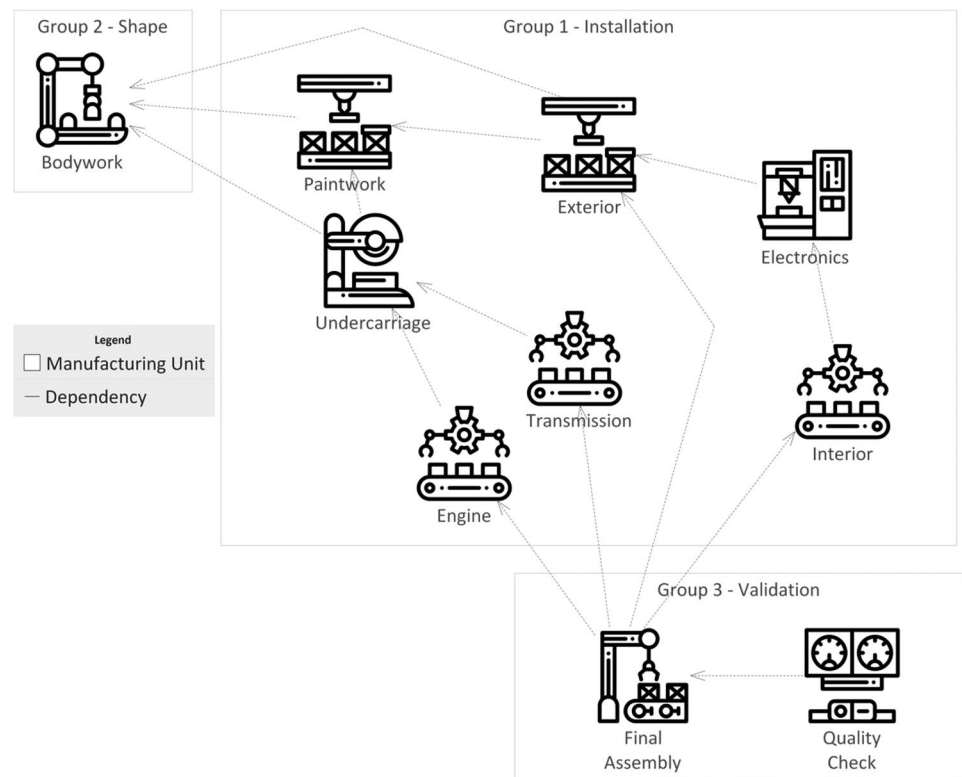


Fig. 17 Dependencies among vehicle manufacturing entities



deals with the functions at the second granularity level. It is therefore essential to manufacture the body before other installations are carried out. The transmission and the engine, on the other hand, require the undercarriage of the vehicle, while the electronic components can only be connected to one another after the exterior has been finalized. For the final assembly, the installation of the engine, transmission, exterior and interior is necessary. The final assembly and the quality check are part of the Validation group, while the body is assigned to the Form group. The remaining production units can be combined to group the installations. This information is important for later production planning. The actual logical model also belongs to the logical perspective, as illustrated in Fig. 18. This diagram uses a class model to store the additional data for the manufacturing units. The transporter that maneuvers the body between the individual machines needs information about which machines still have to be approached. This is solved with Boolean variables by marking the components already installed with true. For the individual production units, data such as length, width and required gaps are marked. For example, the unit that installs the engine is 32 m long, 9 m wide and a further 8 m on the long side for the final processing of the body including the integrated engine. Regarding the width, an additional meter is required to dispose of the old materials. In this context, the logical model relates to the functions of granularity level 2, as do the other models of this viewpoint. The logical model in relation to the Information Layer can also include other levels of granularity.

5.4 Physical data model

The physical data model mainly deals with the technologies used in the system. Therefore, models are required here that define the used data models and database technologies. Before the data models can be created, however, the needed data standards must first be defined. This allows the technology manager to choose the required standards and pass them on for the implementation or provision of technical solutions. In this case study, the logical data model is therefore used to concretise the relationships between the elements and to enrich them with technological aspects. The data standards can be defined for each model of the Information Layer architecture. To accomplish this, a separate DSL with elements for the respective standards is provided. In this context, the diagram with the dependencies between the production units is expanded by different standards, which can be seen in Fig. 19.

The result of this model should show which different technologies are used in such a system and how they can be represented in the Information Layer. This serves as a template for the subsequent implementation of information architectures. In this case study, the dimensions and specifications of the body are passed on to the subsequent production units using JavaScript Object Notation (JSON). The unit for installing the undercarriage only forwards to the engine unit and the gear unit when this has taken place. This can be passed on using an OPCUA protocol. Since the installations for the exterior, the electronics and the interior are different from vehicle to vehicle and are

Fig. 18 Logical model of the case study at level 1

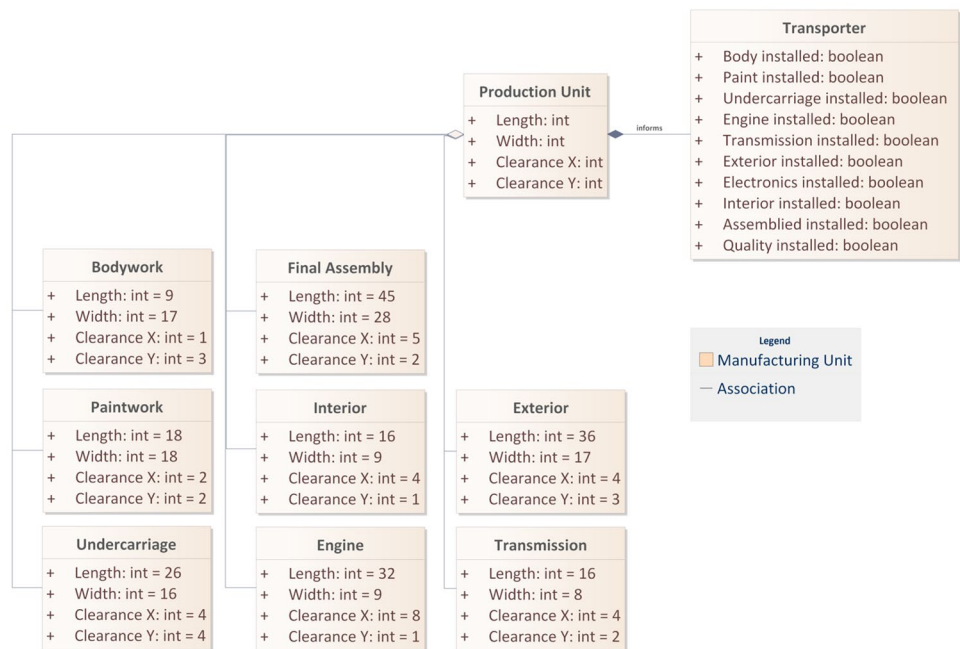
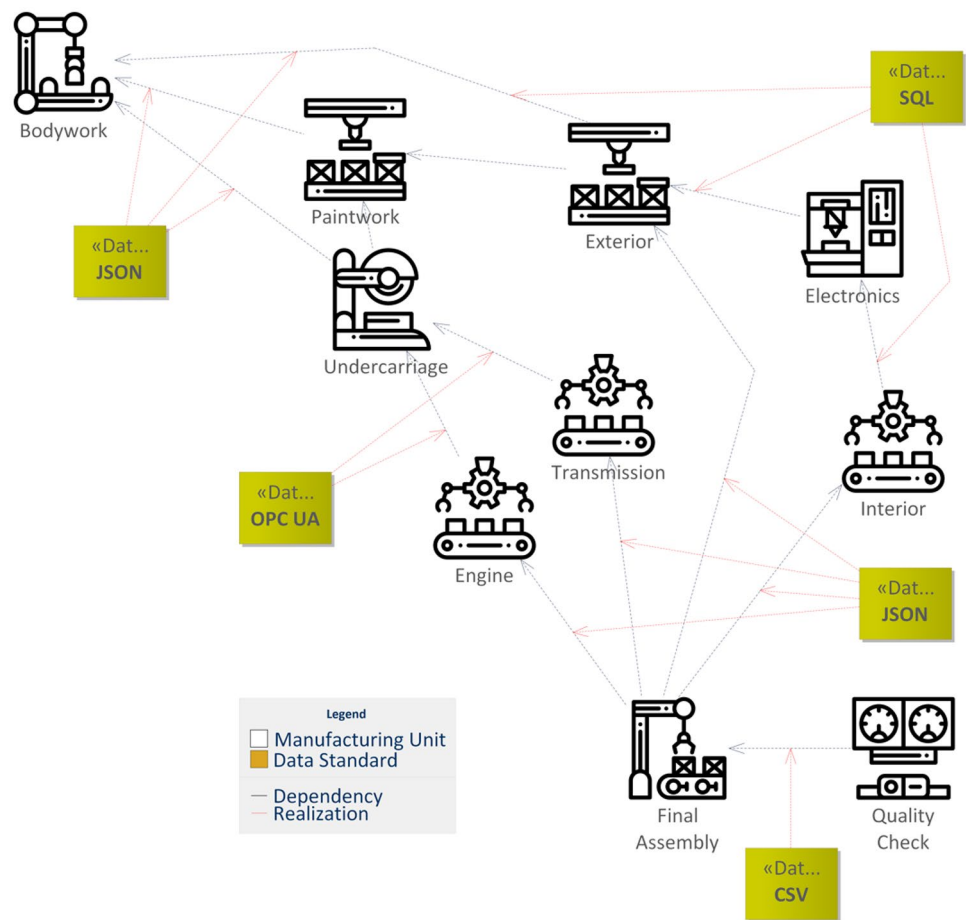


Fig. 19 Data standards in the production process



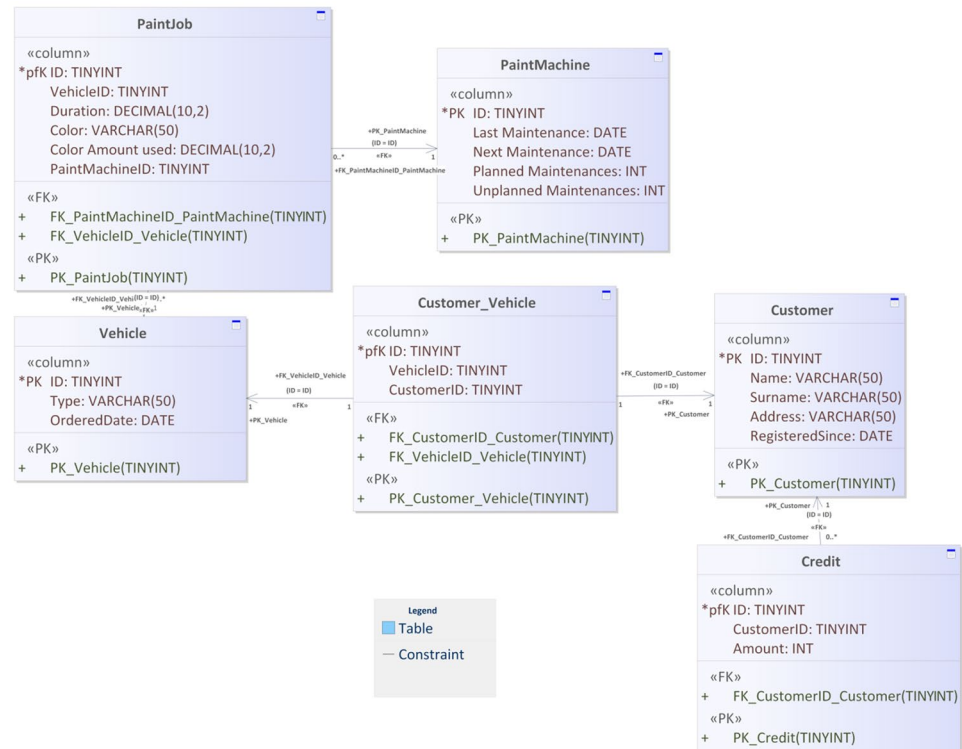
attached individually for each customer, these specifications must be read out by a central unit for the respective vehicle. In this case this is realized by an Structured Query Language (SQL) database, which provides the required data. This database is also dealt with in this viewpoint in the following. The information required for final assembly is then also transferred using an JSON object. This object contains information about which installations the vehicle has and which must therefore be taken into account during final assembly.

Finally, the quality check is informed via a user interface which aspects of the vehicle produced need to be checked. This information is transferred via a CSV file. Another important aspect of the physical perspective is the modeling of the technical environment and the platform for the databases to be implemented. All required database servers, the database management system, the respective resources and required databases are described here. In addition, the interfaces between the individual components are shown. On the basis of this model, the technology manager can determine the required components that the technology operator uses to create the physical system.

In the final model of the physical data model, particular emphasis is placed on the databases themselves. In Fig. 20, an SQL database is shown, which takes previously modeled aspects into account. The entity-relationship diagram of granularity level 1 and the logical data model of granularity level 2 are discussed in particular. The center of the database is the vehicle and the associated customer, which are each saved as a table. These are linked to one another via an additional table based on their respective primary key. This enables a customer to purchase multiple vehicles or a vehicle to belong to multiple customers. Additional information such as name, address or registration date is created for the customer, while the type and date of sale are saved for the car. Regular customers also have the option of applying for a loan. The table for the credit thus has a foreign key to the table of the customers.

A single vehicle has to go through different manufacturing processes before it can be delivered. Using the paintwork as an example, the relationships between the vehicle and the manufacturing process are modeled for the selected database scheme. A vehicle has at least one paint job that is saved in a separate table. This also has information about the duration,

Fig. 20 Example of a table excerpt from the case study database



selected color and amount of this color used. Each individual painting is also carried out on a specific painting machine, which is implemented using an additional external key.

Additional data is also stored for the painting machine, such as information about its maintenance. Through the connection between the painting machine and the painting process, queries such as the number of painting jobs, the average operating minutes per day and the average duration of a painting process can then be implemented. Thus, the physical data model serves as a specification for the actual implementation of the technical concepts, which is described in more detail in the following viewpoint.

5.5 Data definition

The penultimate viewpoint describes the system as it is to be developed. The results from the previous viewpoints are taken up and used to create the system. With regard to the Information Layer, it is important to provide solutions for processing the data. This viewpoint is mainly designed to support the solution provider in their decision-making. For this reason, three different models are created that use granularity levels 2 and 3 of the functional architecture.

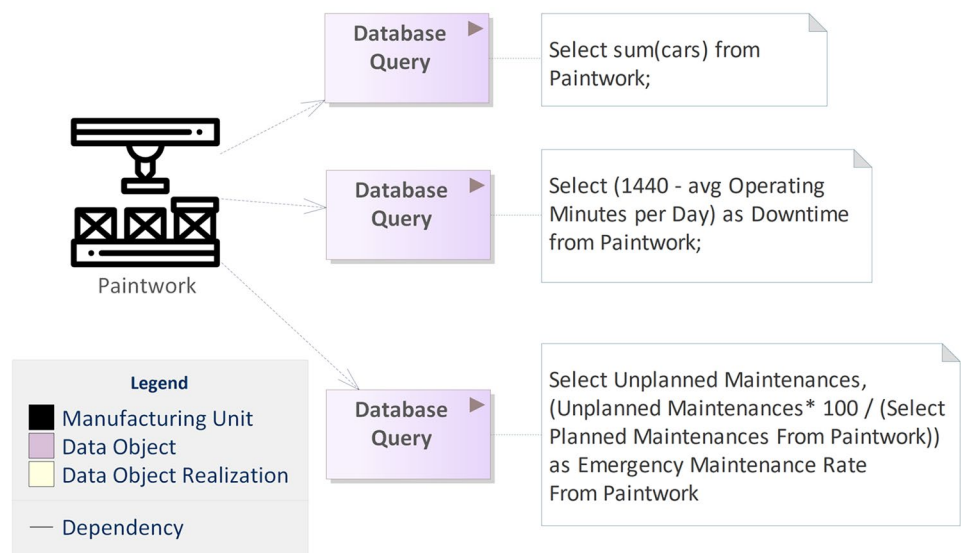
The first model deals with the information objects that arise during the painting process. This enables the programmer to see how he or she can get the information he or she needs. These can then be reused from the model in any application, or even synchronized with the model using

RTE. Database queries are examples of such information objects based on the painting unit. Since the underlying standard is SQL, these are provided in this format. Looking at the data objects in Fig. 21, queries such as the number of painted cars, the downtimes or the emergency maintenance rate can be calculated.

These data could play an important role in maintenance software. However, this model allows far more information than that related to SQL databases. Depending on the data standard, different information can be provided for programming. If, for example, a data object is exchanged via JSON, the structure of the JSON file can be stored in this model. The same principle also applies to data formats based on OPCUA or CSV. The only important thing is to give the programmer the information required to create the application or to generate it automatically from the model.

Looking at the painting process one level below, additional information can also be modeled here, as visible in Fig. 22. In this case study, it is decided for the painting process that a new color is added because a customer has requested this color individually for his vehicle. It can now be documented in the model that a new color and its color code have been added or must be added to the selection. A problem has also been identified with the function that applies the primer on a certain type of vehicle. This is stored in the diagram shown so that the affected stakeholder can take care of the matter. The two models of maintenance show how a diagram can be used to communicate between

Fig. 21 Possible database queries from the database of the painting unit



different stakeholders. In this case, the information exchange takes place beyond the boundaries of the model and it can be used in any diagram of the architecture. However, since an already existing system architecture must exist for this step, it can only take place at the level of the data definition for the providers of solutions for the open improvements or problems.

This viewpoint also allows the data warehouse to be described as a dimensional model with facts and dimensions. As shown in Fig. 23, the exemplary diagram deals with the sales Fact, which inherits three different dimensions. The customer, the vehicle sold and the date of sale represent a single dimension. With the customer, the location is also taken into account, while with the vehicle type, brand and equipment play an important role. Based on these dimensions, different aggregate functions can be made, such as the number of cars sold or the number of customers for a

specific vehicle type. This diagram thus serves as the basis for the subsequent implementation of the data warehouse and informs the implementer about the facts and dimensions to be taken into account.

5.6 Data usage

While the previous viewpoints deal with the system to be created, the system is finally seen as an already implemented and functioning component. This provides the information for the producers of the end product or the consumers of that. For the producer, it is important which data structure the data used have. In the case of an SQL table, for example, it is particularly interesting how many rows and columns are included and which data format has which column. On the basis of this information, the end product can be developed accordingly in order to be able to deal with the data used. On the one hand, the development of interfaces is taken into account, on the other hand, the data storage of the recorded product data is regarded, which is a property of Industry 4.0-capable elements. If, for example, a new machine is developed in this case study that communicates using OPCUA, the required technologies can be integrated using this model.

In addition, local data storage on the machine itself can save certain production data locally based on the specified format. In this image, the database modeled in Fig. 20 is used for the painting process. This means that for this case study this table has the columns ID, vehicle ID, duration of the painting process, selected color, amount of color used and the ID of the painting machine. In order to pass on the data for production, sample data are stored in the model shown. For example, the ID 2 painting process painted the ID 17 vehicle and required just under 2 h, i.e., 118 min. The

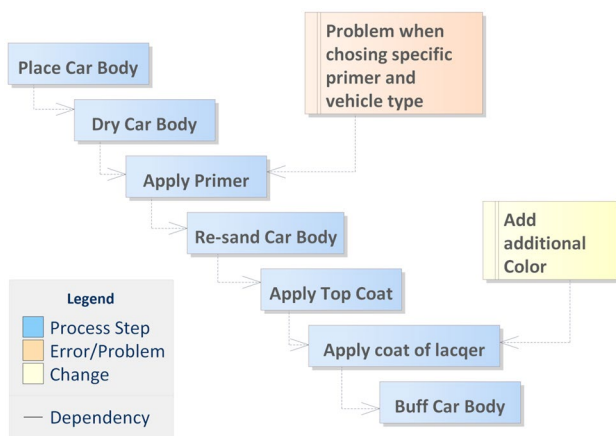
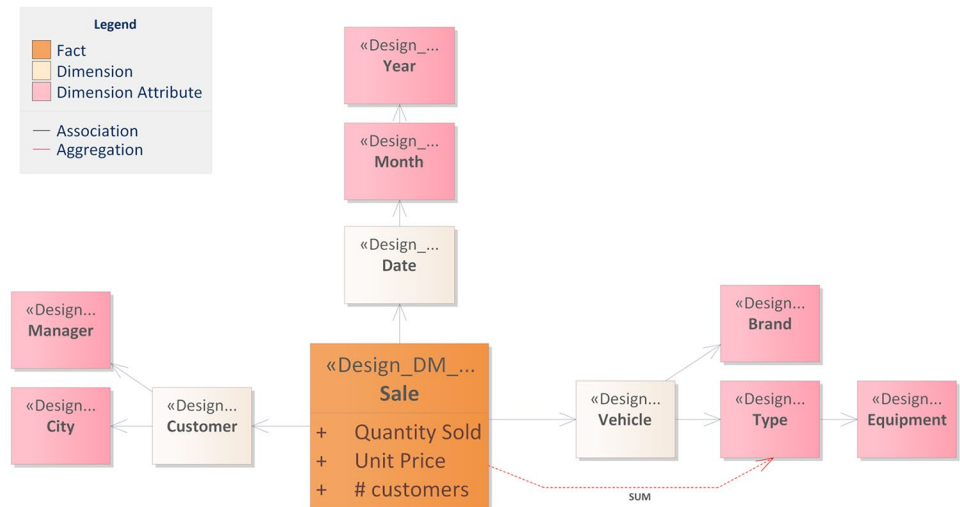


Fig. 22 Potential for change to the painting unit of the case study

Fig. 23 Dimensions-Facts Model of Sales

vehicle was painted blue and 10.7 L of paint were used. All painting processes took place on the same painting machine.

On the other hand, the consumption of the data is another important aspect of the model. This allows the data from the previously modeled system components to be used. In this case study, for example, the data from the painting unit can be used to feed an algorithm for predictive maintenance. The aim of this model is therefore to represent all data that are related to the system in any way. However, since the modeling of all data is too extensive for the model and the database structure of EA is not designed for such an application, the external links to the data must be described in the model itself. Different database connections, storage locations of data in the file system or links to online data can be stored within the diagram. As a result, different consumers can use the respective data according to their interests and access them from the model. This interface satisfies the interests of the consumer by providing access to the actual data sets. It is therefore modeled, how these connections between the model and the stored data within the file system can be implemented by using modeling elements. The painting unit has its own database, which can be accessed via an Open Database Connectivity (ODBC) connection. In addition, a data record with the data from the painting machine is stored directly in the model. However, this is only possible for manageable data. If a certain size is reached, the relationship to the physical data is to be implemented by means of a connection element prepared for this purpose. On the basis of this model, it is now possible to provide the stakeholders with the data they need for a wide variety of applications.

After the information exchange for the production system has been defined at different granularity levels, the next step of MBSE is the specification of technologies and protocols for transferring the information. For this purpose, the entire Information and Communication Technology (ICT)

infrastructure of the system is usually modeled in the Communication Layer. Subsequently, based on these technologies, the interfaces of the respective system elements are now elaborated using MBSE in order to enable the ubiquitous exchange of data. The creation of the associated infrastructure and the technical aspects of the system are then carried out in the next steps of the system development based on RAMI 4.0.

6 Scenario application and evaluation

The chapters described so far illustrate the development of the Information Layer architecture and its application. A wide variety of aspects were addressed in order to address stakeholders and their concerns. The resulting architecture, which is based on ISO 42010, therefore provides a comprehensive concept for creating the information aspect of systems based on RAMI 4.0. Previously it was specified that the Information Layer must support the MBSE and should prepare the information for its application. In this section, it is evaluated, to what extent the approach developed in this work is able to deal with the mentioned aspects. The well-known method for analyzing created software architectures, better known under the term SAAM, is used. As this methodology specifies, different scenarios are generated for the application of the architecture and it is investigated whether it can be used for the individual purposes. Typical Industry 4.0-based scenarios are selected for this, as the purpose of RAMI 4.0 is to create contemporary and future industrial systems. In the following, therefore, each of the respective scenarios is discussed and how these can be addressed by the Information Layer. It specifically deals with how the Information Layer makes the information available so that it can serve and support its application. The application itself and

the subsequent optimization of the resulting data from each algorithm go beyond the aim of this work.

6.1 Industry 4.0 transformation

The first scenario describes an important aspect of Industry 4.0, namely how the Information Layer can contribute to the transformation of the production system according to the aspects of Industry 4.0. It is therefore important to consider this topic in the reference architecture and to provide a suitable application option. In order to implement this abstract scenario concretely and thereby offer the possibility of evaluation with SAAM, the layout optimization of a factory is examined in detail. While linear processes were used in conventional production lines, the layout here, taking the case study into account, should support dynamic production processes and promote the optimized arrangement of the respective production units. In the simplest case, the desired information exchange and the dependencies between the individual units are modeled in the Information Layer, for example in Figs. 16 and 17. The information required for the optimization algorithm is then stored so that it can then be exported from the model. This could be implemented in EA via the RAMI 4.0 Toolbox and its interface. On the basis of the given information and using methods in the area of machine learning, the layout optimization algorithm ideally achieves a better result than the one that is already modeled. It is now the task of the system architect to adapt the model of the information architecture on the basis of this result and to prepare it for the further steps of systems engineering such as the development of the communication infrastructure. In this case, the Information Layer offers a suitable platform for system optimization in the context of Industry 4.0. How this example can be implemented and applied is explained below.

Azadivar and Wang [42] have developed an algorithm that optimizes the layout of machines within a factory. To do this, they use the concepts of simulation and genetic algorithms. In their work they use a simplified factory with rectangular workstations that can only process one part at a time. The transporter itself also only carries a part and the

tasks of the processing sequence remain constant. Using the reverse Polish notation, the layout of the factory is then restored after it has been optimized by means of crossing, mutation and selection. Eight different workplaces were generated as test data for the application, each with different dimensions and spaces. In addition, the transporter has a speed of 10 m per minute, there are four parts to be produced and the routes between the machines are also constant. The cycle time, which is just under 941 min, is determined on the basis of this information.

Thus, this work represents the ideal template for the automobile manufacturer's factory. There are ten factories in use there, but two of them are not used in this scenario. The information for the transporter, the parts to be manufactured and the routes remain unchanged from the referenced work. The next step is to prepare the information required for the optimization algorithm. These are stored in the model as attributes of classes which are associated with the respective production units of the factory. After they have been instantiated, the data of these attributes can be exported from the model using the RAMI 4.0 Toolbox. A CSV file is created to generate the format required for the algorithm. This contains a column for the identification, length, width, spaces and area as well as a line for each of the eight production units. Figure 24 shows the data sets used for the genetic algorithm, which is identical to the data within the generated CSV file. The graphic shows that exactly the required data is generated from the model. In this way, any adjustments can be made in the Information Layer and the algorithm can be executed again with this data.

This specific example is only intended to show that it is possible to optimize any possible system. If the reference architecture is used to model a specific system architecture of an industrial system and if the required data is stored in the diagram like in Fig. 18, this algorithm can be used to optimize the layout of the corresponding system. Based on this, the connections of the production units can subsequently be adapted on the basis of the result. The Information Layer serves as the basis for MBSE and as a starting point for the application of Information Engineering

Fig. 24 Comparison of needed and generated production data

| Workstation identification | Length (X-axis) | Width (Y-axis) | Clearance in X-direction | Clearance in Y-direction | Total area required |
|----------------------------|-----------------|----------------|--------------------------|--------------------------|---------------------|
| 1 | 9 | 17 | 1 | 3 | 10 × 20 |
| 2 | 18 | 18 | 2 | 2 | 20 × 20 |
| 3 | 26 | 16 | 4 | 4 | 30 × 20 |
| 4 | 32 | 9 | 8 | 1 | 40 × 10 |
| 5 | 45 | 28 | 5 | 2 | 50 × 30 |
| 6 | 16 | 8 | 4 | 2 | 20 × 10 |
| 7 | 36 | 17 | 4 | 3 | 40 × 20 |
| 8 | 16 | 9 | 4 | 1 | 20 × 10 |

methods and the resulting system transformation according to the aspects of Industry 4.0.

6.2 Process optimization

The second scenario addresses the actual purpose of Industry 4.0. The focus here is on optimizing manual or complex processes. By collecting a lot of data, analyses can be made based on this or processes can be simulated. Data-based optimizations are also possible in the area of machine learning by transferring the results of the algorithms into the system. Here the Information Layer is an important interface that recognizes the potential for optimization, makes the data available for the algorithm and can incorporate the result of the optimization into the architecture of the system. The example to be used for the concrete application of this scenario applies typical production planning. In production planning, a behavior similar to that of the previously mentioned layout optimization is aimed for. The Information Layer should provide a basis for the production process and its model. The modeled information can then be exported with the help of the RAMI 4.0 Toolbox for the optimization algorithm, which achieves a better production process or shows problems in the current process. This result should then be considered again in the model and thus in the architecture of the instantiated manufacturing system, which can then be adapted accordingly in order to subsequently elaborate the communication infrastructure. In this scenario, however, the Information Layer is an important source of information for optimization algorithms in the sense of Information Engineering and provides the data for it. MBSE is also supported by the adaptation and evolutionary development of the system architecture.

In [43] a framework is introduced that provides an integrated simulation environment. Using a learning model, the potential of data-driven reinforcement learning in production planning is examined. This means that a wide range of different production systems can be parameterized and then simulated and should address the complexity of future production systems, which are becoming increasingly more complicated due to production in batch size 1. This is due to different publication dates, sequence-dependent set-up times, prescribed due dates, different process types and irregular downtimes. Therefore, this algorithm helps to support the planning and control of complex manufacturing systems through data-driven reinforcement learning. Different scenarios can be simulated with different options. In this work, for example, the number of machines and their classification are dynamically adopted from the model. The framework itself is implemented in Python and can therefore run in a wide variety of operating systems and under different conditions. Python allows algorithms to be executed with little effort. In order to be able to execute simple optimization

algorithms on the local computer, a Python installation is carried out with all standard libraries. In the context of this work, Python for Windows 10 under version 3.8.5 is used. The flexible setting options allow the analysis of different scenarios. The following parameters can be freely selected in the model:

- Resources: The number of machines, new tasks, storage areas for finished products and transport resources can each be set.
- Layout: The layout of the resources can be fixed using a distance matrix.
- Tasks: The generation process, the buffer capacity and restrictions can be specified for newly created tasks.
- Machines: The machine parameters relate to buffer capacities, processing times, groupings, dismantling processes and changeover times.
- Storage areas: The capacity can be freely selected here.
- Transport resources: Processing times and transport speeds are to be stored here.
- General: In addition, the distribution of tasks, processing times for loading and unloading as well as simulation-specific settings can be made.

The individual parameters can be stored centrally in an initialization file. This data is used during the start of the simulation and the scenario is adjusted accordingly. It is therefore necessary to adapt this file and replace it with parameters from the model. Since the dynamic transfer of each of the named parameters into the simulation environment is beyond the aim of this work, the order of the jobs, the number of machines and the grouping of the machines are taken from the model in this application scenario. The data required for this are stored in the Information Layer in the logical model, as seen in Fig. 17. From there, the attributes of each machine are iterated using the RAMI 4.0 Toolbox and their group membership is extracted. The orders are generated randomly by the code. In order to save the data in the initialization file, it is created as a template. The information to be stored is stored in this template file as uniquely identifiable character strings which, when replaced by the information generated from the model, result in an instantiated initialization file. This is embedded in the Python environment and called the next time the simulation is started. In conclusion, by executing this algorithm, the Information Layer can contribute to the optimization of production processes.

6.3 Stakeholder communication

The last scenario takes care of the communication of the stakeholders in an Industry 4.0-based system. Here the model of the system can capture and take into account the interests of various stakeholders. This can either be done

directly through the modeled diagrams or additional information can be stored in the model. For example, in Fig. 14 the management and the data managers meet, who both have their interests in the system. Therefore, to ensure common communication, both the desired management process and the required data exchange are stored. In this case, both stakeholders are addressed and the system can be developed based on consensus. A similar example is Fig. 19, where the dependencies between the manufacturing units are shown, but also the standards for ensuring the exchange of information. Here it is important that both interests are taken into account and not interfere with one another. The model serves as a means of communication between the stakeholders so that, on the one hand, the data standards are taken into account and, on the other hand, the dependencies are not circumvented. Communication between stakeholders can also take place via metadata in the model. An example of this is the calculation and provision of KPI. In this way, certain stakeholders can store data for the calculation of these KPI in the model and other stakeholders can access this data in order to carry out further analyses. Examples of such key figures are given below, with particular emphasis on Industry 4.0. In this scenario, the Information Layer serves as a communication interface and for data administration, which supports both model-based systems engineering and Information Engineering.

The evaluation of KPI is a constant factor in how companies can measure their success. However, this was a common method even before the fourth industrial evolution. With the upswing of Industry 4.0, new KPI have been created and some previously insignificant KPI have gained in importance. Therefore, special focus is placed on those KPI that can provide information about future production systems. In addition, it must be possible to calculate these KPI from the model. This enables stakeholders to communicate via the model, which is an important aspect of MBSE or reference architectures. This can be done if some stakeholders store information in the model and others receive the results of evaluations of this information. Therefore, the four KPI are examined in this scenario. The result of the KPI calculation is depicted in a simple message window, as can be seen in Fig. 25. This is seen as a suitable possibility for the first iteration of ADSRM. In future scenarios, among other things, access rights and individual output options must be taken into account. The KPI selected in this scenario are explained in more detail below.

6.3.1 Factory employee rate

Since Industry 4.0 attaches particular importance to process automation and drives an optimal use of resources through ubiquitous communication between the machines, the required employees in these factories are steadily reduced.

Number of operators in the factory: 20%
E2E production time: 1140 min
Emergency Maintenance Rate: 12,5%
Downtime: 268 min

OK

Fig. 25 Output window of the KPI calculation

As a result, this metric indicates how many employees in the company actually work in the factories. While the absolute number hardly allows any conclusions to be drawn, the relationship to all employees can be an important factor for the long-term prognosis. In order to calculate this KPI, the number of employees for each department is simply added as a tagged value in the model. This means that all departments can be iterated through, the respective employees can be added up and the relationship to the production department can be calculated. In this example, the company has a total of 580 employees, with 120 working in production, resulting in a factory employee rate of 20%.

6.3.2 End-to-end production time

Another important indicator of the company's success and its performance is the end-to-end production time. This can be used to calculate how long it takes to manufacture products. This metric needs to be undercut in the following so that the company is able to manufacture a larger number of products in the same period of time. While this metric is an important factor for manufacturing in batch size 1, calculating the end-to-end production time from the model is complicated. This is mainly due to the fact that the data from each individual manufacturing process must be used. Therefore, in this scenario, the mean value of the production times of each machine is used, which in turn is stored in the individual model elements in Fig. 16. Similar to the calculation of the factory employee rate, the individual times are searched for from the elements, added up and thus summed up to give the end-to-end production time. In this case, the production time of a car takes 1140 min, which means 19 h.

6.3.3 Emergency maintenance rate and downtime

While maintenance cycles can be optimized by storing a large amount of data, both of these metrics are gaining in

importance. By precisely predicting upcoming maintenance, the emergency maintenance rate should be as low as possible. The same applies to machine or production downtime. Since production comes to a standstill during unexpected maintenance, this increases downtime. Thus the two KPI are related. The emergency maintenance rate is calculated by comparing all maintenance carried out with the unplanned ones, which in this case results in a value of 12.5%. The downtime is added up over the individual downtimes and returns the production downtime over a certain period of time. In this scenario, this is 268 min, that means just under 5 h for the selected period. This information can be generated by database queries and stored and communicated via the model in Fig. 21.

Communication within an Industry 4.0 system can be improved through the above-mentioned diagrams and the calculation and evaluation of KPI. In this way, stakeholders who are interested in certain parts of the system can be addressed directly and without detours. The reference architecture of the Information Layer plays an important role here, as it stores the most important data for each instantiated system and makes it available via the RAMI 4.0 Toolbox. This means that both industrial systems engineering and Information Engineering are supported in this area. However, it must be taken into account here that the respective modeled system and its architecture are evolutionarily adapted and constantly brought up to date, as it is in the sense of the digital twin.

6.4 Evaluation results and discussion

The application scenarios described in this section are intended to evaluate whether it is possible to display all required information in the Information Layer of RAMI 4.0 in such a way that it can be used from the model for Information Engineering. Since RAMI 4.0 mainly supports industrial production in the context of Industry 4.0 and the associated systems engineering, the ability of the developed architecture to implement use cases in this area must be evaluated. The analysis method SAAM should therefore examine whether the reference architecture of the Information Layer is able to integrate these typical scenarios, which lead back to the chosen case study. By utilizing SAAM, a comparative analysis is conducted, based on the respective application scenarios. Thereby, the application value of the Information Layer architecture is investigated based on typical industrial utilization examples. Special focus is thereby set on the chosen quality attributes, feasibility, usability, integrity and availability. By applying this evaluation methodology based on SAAM, those attributes are compared with each other with regard to multiple application possibilities. This means, different scenarios are investigated, each of them exhibiting different results. The comparison of those results

and the subsequent weighting indicates degrees of fulfillment and significance of each quality attribute concerning the application value of the proposed reference architecture.

By doing so, the first application scenario describes the transformation of the production system according to the concepts of Industry 4.0. This is addressed by several models in the reference architecture. New ideas for automation or the information potential of Industry 4.0-compliant data acquisition can be modeled as shown in Fig. 13. Another example shows the potential for modifications in the system and thus takes Industry 4.0 aspects into account. In addition, the example of layout optimization is a further possibility to support systems engineering and to transform the industrial system towards Industry 4.0 on the basis of the system architecture. In summary, it can be said that the first application scenario is addressed in several places in the proposed reference architecture and offers extensive possibilities to integrate the concepts of Industry 4.0 into an industrial system. In the second scenario, process optimization and the acquisition of more valuable information are in the main focus. The associated manufacturing processes or the provision of information for their analysis takes place at various points in the model, as outlined in the section. The respective results are then used for further processing or flow back into the model of the system architecture. The last specified scenario is about communication between the system stakeholders. Some examples are given of how an instantiated system architecture based on the developed reference architecture can serve as an interface between the stakeholders.

Even if the Information Layer itself still offers several options for implementing the application scenarios, the selected examples are primarily designed as a first-time Proof-of-Concept (PoC) and should demonstrate a possible implementation. It can thus be said that the reference architecture of the Information Layer is able to enable the application of the individual scenarios in the context of the case study. Due to the fictitious origin and the superficiality of the application scenarios, this work should primarily demonstrate the feasibility of how a relationship between model and application can look. This feasibility study generates an idea of what is possible in this area. The actual evaluation of the results for realistic industrial purposes goes beyond the aim of the thesis and should be examined with a more extensive case study. The case study used in this work should rather check the integrity of the models as well as their availability for problems. It is particularly important to note whether each concern is addressed by one or more models and whether these models are consistent with one another. Therefore, the following paragraphs describe how the quality attributes to be evaluated, usability, feasibility, integrity and availability, are addressed and fulfilled by the three application scenarios, which are shown in Table 4.

Table 4 System quality attributes and their degree of fulfillment

| Quality Attribute | Transformation | Optimization | Communication | Weighting | Mean | Weighted Fulfillment |
|-------------------|----------------|--------------|---------------|---------------|------|----------------------|
| Feasibility | 45% | 35% | 100% | $\frac{1}{2}$ | 60% | 30% |
| Usability | 80% | 60% | 100% | $\frac{1}{3}$ | 80% | 26,67% |
| Integrity | 30% | 90% | 30% | $\frac{1}{6}$ | 50% | 8,33% |
| Availability | 90% | 70% | 20% | $\frac{1}{9}$ | 60% | 6,67% |

The evaluation of the feasibility was particularly emphasized and represented a great effort in this work. Therefore, the evaluation was weighed as much as all other quality attributes combined, which resulted in being the most fulfilled attribute in concern to the weighted degree of fulfillment. A difficult undertaking here was the search for actual concrete applications for the respective scenarios. Since the aim of this work was not to create a new algorithm, but rather to use existing data, suitable frameworks had to be used here. However, although many of the terms mentioned represent important concepts in the context of Industry 4.0 to enable this change, the actual implementation is still at the beginning of research. Above all, this may be due to the fact that industrial information engineering in the area of Industry 4.0 also needs further research. However, some application examples have recently been published, as this section has shown. The result of the analysis of the architecture and the application of its information by SAAM shows that the Information Layer can provide the data required for the implementation of the scenarios, in particular with regard to the stakeholder communication. In addition, the reference architecture considering the case study addresses the interests of all twelve stakeholders and provides the information required for each of the three application scenarios, which guarantees feasibility in the context of this work. Nevertheless, both Industry 4.0 Transformation as well as Process Optimization needed a significant amount of workaround to be applied in this evaluation strategy, which resulted in giving away points concerning the feasibility. Thus, a more realistic example with an actual industrial origin should evaluate the architecture developed in this thesis on the basis of new requirements. As a result, its suitability for future system development in the context of RAMI 4.0 can be validated with more extensive case studies, which must be implemented in the next iterations of ADSRM.

With the development of the RAMI 4.0 Toolbox, however, the usability has been increased considerably by exporting data at the push of a button and automatically initializing external algorithms. Therefore, without factoring in the weighting, it reaches a degree of fulfillment of 80% and almost as much as the feasibility including the weighting. In relation to the three application scenarios, at least one specific application was implemented for each

individual scenario and called automatically from the case study system architecture using the RAMI 4.0 Toolbox. Using the selected scenarios, this is intended to demonstrate, for example, how manual or constantly repetitive tasks can be simplified and partially automated through the development of functions. This ensures and increases the usability in this work. Another aspect regarding the usability in this thesis is the use of a DSL. In this way, certain stakeholders can use the modeling language they are familiar with and do not have to learn any new technologies or languages. Examples of this are the integration of simple symbols for the machines, the use of data flow diagrams for modeling the data exchange or modeling the tables in the database for its real implementation. The integrity of the model is ensured in that no information is redundant or the same information is not modeled in different diagrams. For this reason, the same concepts are shown differently in different diagrams, so that the known notation can be made available to the respective stakeholder. In particular, the Process Optimization inherits a large integrity, as the information is solely stored within one specific model. The other application scenarios indicated the possibility for data manipulations, which resulted in a degree of fulfillment of only 50%. Availability is about whether a suitable diagram is offered in the model for each interest of each stakeholder so that the respective interests can be satisfied. Similar to the integrity, this quality feature is difficult to validate in the present work. However, as stakeholder communications needs a lot of different diagrams, only 20% have been reached, as the logical model for the Industry 4.0 Transformation is created anyways, thus gaining 90%. On the basis of the case study selected in this work, the model is consistent and diagrams for each stakeholder concern are available. This means that the integrity and availability are ensured in this work, but only make up less than 10% respectively when considering the weighted degree of fulfillment. The assertion whether this also applies to every other industrial system that is modeled on the basis of the reference architecture for the Information Layer can hardly be made with the information available. Here, too, in the next iterations of ADSRM, more extensive case studies from other industries must be implemented with the result of this work. In summary, based on the evaluation according to SAAM, it can be said that the most important quality

attributes, feasibility as well as usability, are fulfilled to a large extent by the Information Layer reference architecture, while there remains optimization potential considering integrity and availability. Thus, the introduced architecture is ready-to-use for application in more comprehensive examples, while the application value could be enhanced. This needs to be done in this work's prospect, which is delineated within the next section.

7 Conclusion and future work

The term Industry 4.0 refers to the transformation from conventional production systems to value creation networks that ubiquitous communicate with one another. This is mainly due to new technologies in the area of IIoT or CPS. The ubiquitous communication can automate processes and optimize resource consumption. Another factor in this network is the increasing amount of data being collected. More and more machine-, process-, manufacturing- and metadata are accumulating. Under the term Big Data, this data should be stored sustainably and processed for other purposes. This increasing amount of data and automation possibilities in production systems result in a higher complexity compared to conventional production systems. In order to counteract this complexity and to enable the production of such systems while taking all aspects into account, the three-dimensional reference architecture model RAMI 4.0 was developed by several German associations. Its application is hardly specified, as this model is mainly theoretically described. Considering the just mentioned aspects, the aim of this paper is to enable the application of the RAMI 4.0 Information Layer in order to support system development and to ensure data application in terms of Information Engineering. At first, therefore, a reference architecture is developed in Sect. 4, which refines the aspects of RAMI 4.0 and includes a detailed addressing of all stakeholders. This is carried out in smaller steps, taking into account the agile development method ADSRM and the case study of a modular automobile production, whereby the focus is primarily on an applicable solution. The reference architecture itself must support MBSE of industrial manufacturing systems and enable the use of the data for optimization algorithms from information engineering. The extent to which the resulting reference architecture succeeds in doing this is evaluated on the basis of the analysis method SAAM in Sect. 6. The result of this work is therefore an applicable reference architecture of the Information Layer of RAMI 4.0, which supports the creation of industrial systems and provides Information Engineering algorithms with suitable data.

The result of this work is by no means a completely ready-to-use method, but a first step in the right direction. The present work mainly examines the feasibility of a reference architecture for an information model. Therefore,

additional research topics should be pursued based on this result in subsequent projects. Actual case studies from industry must be used to validate whether the solution achieved is applicable in different industries. In this way, the reference architecture of the Information Layer can be refined and the necessary adjustments can be made. This should be done by defining new requirements, developing new artifacts and implementing projects based on the selected case studies in the next iterations of ADSRM. A next follow-up project is the refinement of the interface to the methods from Information Engineering. Here, not only a few data should be transferred from the model, but the complete parameters required for the respective algorithm should be transferred. Starting an external program with a single click would improve the usability and dynamic optimization of individual industrial systems. In addition, new optimization algorithms must constantly be taken into account and implemented in order to keep an eye on the state of the art and to remain future-oriented. Finally, it is necessary to integrate the Information Layer into the remainder of the architecture of RAMI 4.0 in order to enable comprehensive system engineering. Above all, the aspects of the adjacent layers, i.e., the functions from the Function Layer and the technologies from the Communication Layer, should be taken into account here. This aims to counteract with the emerging complexity of current and future production systems and deal as a basis for joint system engineering across several industries and addressing all stakeholders.

Acknowledgements 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.

Funding Open access funding provided by FH Salzburg - University of Applied Sciences. The work has been funded by the Christian Doppler Research Association.

Availability of data and materials Not applicable

Code availability The RAMI 4.0 Toolbox with all implemented functions is available is available at <https://rami-toolbox.org/download/>.

Declarations

Ethics approval Not applicable

Consent to participate All authors understand that they will not benefit directly from participating in this research. They understand that all information provided for this study will be treated confidentially.

Consent for publication All authors have read and agreed to the submitted version of the manuscript and give our consent to be published in *The International Journal of Advanced Manufacturing Technology*.

Conflicts of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- He Z, Xu J, Tran KP et al (2021) Modeling of textile manufacturing processes using intelligent techniques: a review. *J Adv Manuf Technol* 116(1):39–67
- Rajkumar R, Lee I, Sha L et al (2010) Cyber-physical systems: the next computing revolution. In: Design automation conference, IEEE, pp 731–736
- Claude Jr RM, Horne DA (1992) Restructuring towards a service orientation: the strategic challenges. *Int J Serv Ind* 3(1):0–0
- Frank AG, Dalenogare LS, Ayala NF (2019) Industry 4.0 technologies: Implementation patterns in manufacturing companies. *Int J Prod Econ* 210:15–26
- Haberfellner R, de Weck, O, Fricke E et al (2015) Systems Engineering - Grundlagen und Anwendung, 13th edn. Orell Füssli
- DeLaurentis D (2005) Understanding transportation as a system-of-systems design problem. In: 43rd AIAA Aerospace Sciences Meeting and Exhibit, p 123
- Friedenthal S, Oster C (2017) Architecting Spacecraft with SysML: A Model-Based Systems Engineering Approach. CreateSpace Independent Publishing Platform
- Jones D, Snider C, Nassehi A et al (2020) Characterising the digital twin: A systematic literature review. *CIRP J Manuf Sci Technol* 29:36–52
- Kronberger G, Bachinger F, Affenzeller M (2020) Smart manufacturing and continuous improvement and adaptation of predictive models. *Procedia Manufacturing* 42:528–531
- Wan L, Chen H, Ouyang L et al (2020) A new ensemble modeling approach for reliability-based design optimization of flexure-based bridge-type amplification mechanisms. *J Adv Manuf Technol* 106(1):47–63
- Hankel M, Rexroth B (2015) The Reference Architectural Model Industrie 4.0 (RAMI 4.0). ZVEI
- Cruz Salazar LA, Ryashentseva D, Lüder A et al (2019) Cyber-physical production systems architecture based on multi-agent's design pattern—comparison of selected approaches mapping four agent patterns. *J Adv Manuf Technol* 105(9):4005–4034
- Deutsches Institut für Normung (2016) DIN SPEC 91345: 2016–04. Reference Architecture Model Industrie 4.0
- International Electrotechnical Commission (2016) IEC 62890: Life-cycle management for systems and products used in industrial-process measurement, control and automation
- International Electrotechnical Commission (2016) IEC 62264: Enterprise-control system integration
- International Electrotechnical Commission (2001) IEC 61512: Batch control
- Bitkom, VDMA, ZVEI (2015) Umsetzungsstrategie Industrie 4.0, Ergebnisbericht der Plattform Industrie 4.0. ZVEI
- Wymore AW (2018) Model-based systems engineering, vol 3. CRC Press
- Weilkiens T (2011) Systems engineering with SysML/UML: modeling, analysis, design. Elsevier
- Weilkiens T, Lamm JG, Roth S et al (2015) Model-based system architecture. John Wiley & Sons
- Brambilla M, Cabot J, Wimmer M (2017) Model-driven software engineering in practice. *Synthesis Lectures on Software Engineering* 3(1):1–207
- Heilmann H, Heinrich LJ, Roithmayr F (2018) Information Engineering: Wirtschaftsinformatik im Schnittpunkt von Wirtschafts-, Sozial- und Ingenieurwissenschaften. Walter de Gruyter GmbH & Co KG
- Martin J (1989) Information engineering: introduction. Prentice Hall PTR
- Wirth R, Hipp J (2000) Crisp-dm: Towards a standard process model for data mining. In: Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining, Springer-Verlag London, UK
- Grangel-González I, Halilaj L, Coskun G et al (2016) Towards a semantic administrative shell for industry 4.0 components. In: 2016 IEEE Tenth International Conference on Semantic Computing (ICSC), IEEE, pp 230–237
- Fischbach A, Strohschein J, Bunte A et al (2020) Caai—a cognitive architecture to introduce artificial intelligence in cyber-physical production systems. *J Adv Manuf Technol* 111(1):609–626
- Zhang X, Ming X, Yin D (2019) Reference architecture of common service platform for industrial big data (i-bd) based on multi-party co-construction. *J. Adv. Manuf. Technol.* 105(5), 1949–1965
- Arantes M, Bonnard R, Mattei AP et al (2018) General architecture for data analysis in industry 4.0 using sysml and model based system engineering. In: 2018 Annual IEEE International Systems Conference (SysCon), IEEE, pp 1–6
- Pisching MA, Pessoa MA, Junqueira F et al (2018) An architecture based on rami 4.0 to discover equipment to process operations required by products. *Comput Ind Eng* 125:574–591
- Sharpe R, van Lopik K, Neal A et al (2019) An industrial evaluation of an industry 4.0 reference architecture demonstrating the need for the inclusion of security and human components. *Comput Ind* 108:37–44
- Mandel C, Stürmlinger T, Yue C et al (2020) Model-based systems engineering approaches for the integrated development of product and production systems in the context of industry 4.0. In: 2020 IEEE International Systems Conference (SysCon), IEEE, pp 1–7
- Wortmann A, Barais O, Combemale B et al (2020) Modeling languages in industry 4.0: an extended systematic mapping study. *Softw Syst Model* 19(1):67–94
- Hevner A, Chatterjee S (2010) Design science research in information systems. In: Design Research in Information Systems. Springer, p 9–22
- Conboy K, Gleasure R, Cullina E (2015) Agile design science research. In: International Conference on Design Science Research in Information Systems, Springer, pp 168–180
- Kazman R, Abowd G, Bass L et al (1996) Scenario-based analysis of software architecture. *IEEE software* 13(6):47–55
- Walzl H, Wildemann H (2014) Modularisierung der Produktion in der Automobilindustrie. TCW, Transfer-Zentrum
- Walzl H, Wildemann H (2014) Der modulare produktionsbaukasten teil 2. *Productivity Management* 19(4):41–44
- Antunes G, Barateiro J, Becker C et al (2011) Modeling contextual concerns in enterprise architecture. In: 2011 IEEE 15th International Enterprise Distributed Object Computing Conference Workshops, IEEE, pp 3–10
- Ballejos LC, Gonnet SM, Montagna JM (2008) A stakeholder model for interorganizational information systems. In: International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, pp 73–87

40. Binder C, Polanec K, Brankovic B et al (2021) Enabling model-based requirements engineering in a complex industrial system of systems environment. In: 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2021). IEEE
41. Binder C, Draxler D, Neureiter C et al (2019) Towards a model-centric approach for developing functional architectures in industry 4.0 systems. In: 5th IEEE International Symposium on Systems Engineering (ISSE 2019). IEEE
42. Azadivar F, Wang J (2000) Facility layout optimization using simulation and genetic algorithms. *Int J Prod Res* 38(17):4369–4383
43. Kuhnle A, Kaiser JP, TheiSS F et al (2021) Designing an adaptive production control system using reinforcement learning. *J Intell Manuf* 32. <https://doi.org/10.1007/s10845-020-01612-y>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.