

Article

Manufacturing Execution Systems for the Food and Beverage Industry: A Model-Driven Approach

Xinyu Chen *, Christoph Nophut and Tobias Voigt

Chair of Food Packaging Technology, TUM School of Life Sciences, Technical University of Munich, 85354 Freising, Germany; christoph.nophut@tum.de (C.N.); tobias.voigt@tum.de (T.V.)

* Correspondence: xinyu.chen@tum.de

Received: 8 November 2020; Accepted: 28 November 2020; Published: 2 December 2020



Abstract: Manufacturing Execution Systems (MES) are process-oriented information-technology (IT) solutions for collecting and managing information from manufacturing processes. Due to the individual programming effort and the complex integration with other manufacturing systems, though the food and beverage manufacturers can benefit from the MES, its implementation is not widespread in this industry. To simplify the implementation and engineering process, the concept of model-driven engineering (MDE) is considered as a solution. However, a feasible model-driven approach for MES engineering has not been established, not to mention for the food and beverage industry. This paper presents an approach for the automatic MES generation according to the MDE concept providing MES functions that are relevant to the food and beverage manufacturing processes primarily. It consists of necessary phases to cover the whole engineering process of the MES. Based on the application of the presented approach to the brewing process in a brewhouse, the feasibility and practicality of this approach were proven.

Keywords: manufacturing execution systems; model-driven engineering; food and beverage industry

1. Introduction

Information-technology (IT) systems have been used to improve production processes and are indispensable for manufacturing firms seeking to remain competitive in the global market [1]. To utilize the potential for improvement, manufacturing processes on the shop floor should not be ignored. With the increasing flexibility of processes and the diversification of products, the traditional operational IT systems (e.g., Enterprise Resource Planning (ERP)), are not able to react in real-time to meet the requirements of improvements on the shop floor. Manufacturing Execution Systems (MES) are process-oriented software systems that manage and analyze real-time information in the manufacturing processes. On one hand, MES implement production plans from systems on the enterprise-level in operative details to the production area. On the other hand, they provide essential key performance indicators, including specific energy consumption and machine efficiency metrics, to the enterprise for making business decisions and for the improvement of the manufacturing performance. The implementation of MES raises the transparency of manufacturing processes so that product quality and workflow efficiency can be optimized. Although the sectors in the food and beverage industry can benefit from MES, the special characteristics of the manufacturing processes and the lack of human, material, and financial resources have hindered their investment into MES implementation, in which considered efforts for programming, customizing, and integrating are indispensable to deal with the high complexity of MES, owing to the wide range of functions required to control the shop floor processes and the communications with intra- and extra-systems [2,3].

Using code-centric technologies to develop such a complex system requires a herculean effort [4]. An essential factor behind the development is the wide conceptual gap between the implementation

domains. The concept of model-driven engineering (MDE) is concerned primarily with reducing this gap through the use of abstractions to software implementations [5,6]. For the software development using MDE concept, the primary artifacts of development are models, the transformation from models into running systems is relied on computer-based technologies. By raising the level of abstraction and automating labor-intensive and error-prone tasks, MDE has been seen as a solution to handle the complexity of software development [7]. However, though many researchers have focused on the industrial application of the MDE, such as for embedded software [8], mobile applications [9], and the automated production systems [10], a concrete approach that covers the whole lifecycle of the MES engineering has not been established yet.

This paper presents the first model-driven approach for the automatic generation of the MES and emphasizes specification as well as MES generation with validation to fulfill the requirements from the food and beverage industry. The feasibility of this approach is proven by its application to a brewing process for thermal energy management. The paper is structured as follows. Section 2 introduces the definition and functionality of the MES and gives a literature review to analyze the requirements and barriers from the food and beverage industry for MES implementation. Previous works related to MDE have also been presented in this section. Section 3 presents the requirements that should be fulfilled by a model-driven approach for MES engineering based on the literature review. Section 4 describes the developed model-driven approach in detail. Section 5 provides a use case on the application of this approach to a brewhouse and evaluates it according to the requirements presented in Section 3. Section 6 draws conclusions and outlines future work.

2. State of the Art

2.1. Definition and Functionality of the MES

For the manufacturing industry, to fulfill the requirement from the shop floor on the data collection and operation reaction in real-time, the idea of MES was firstly introduced by AMR Research in 1992, and also in this year, MESA (Manufacturing Enterprise Solutions Association) was established [11,12]. According to MESA, the core task of the MES is considered as an information center that enables the manufacturers to optimize production activities [13]. For different manufacturing environments, twelve MES functionalities were defined [14]: resource allocation and control, dispatching production, data collection and acquisition, quality management, process management, production tracking, performance analysis, operations and detailed scheduling, document control, labor management, maintenance management, and transport, storage, and tracking of materials.

According to hierarchy model defined in ISA-95 [15], the MES are located on level 3 and act as the bridge between the automation layer and the enterprise layer [16]. Due to the development of automation and information technology, the idea of Industry 4.0 was officially announced in 2013 as a German strategic initiative to keep a pioneering role in the revolutionizing industries [17]. Three dimensions are contained in the paradigm of Industry 4.0: horizontal integration across the entire value creation network, vertical integration, and networked manufacturing systems; end-to-end engineering across the entire life cycle [18,19]. A three-dimensional model named RAMI 4.0 is proposed as the reference architecture model bringing the essential elements together with different standards in order to implement Industry 4.0 techniques (Figure 1) [20].

RAMI 4.0 supplements the hierarchy level defined in IEC 62,264 with a lower level of “Product” and an upper level of “Connected World”, which integrates the control systems and establishes the environment of Industry 4.0. The left horizontal axis represents the life cycle of systems and products based on IEC 62890. With the six layers on the vertical axis, the components of Industry 4.0 are described structurally [21]. As the center for collecting, processing, and delivering the information both in the horizontal and vertical direction, the MES is indispensable in the developing manufacturing industries [22].

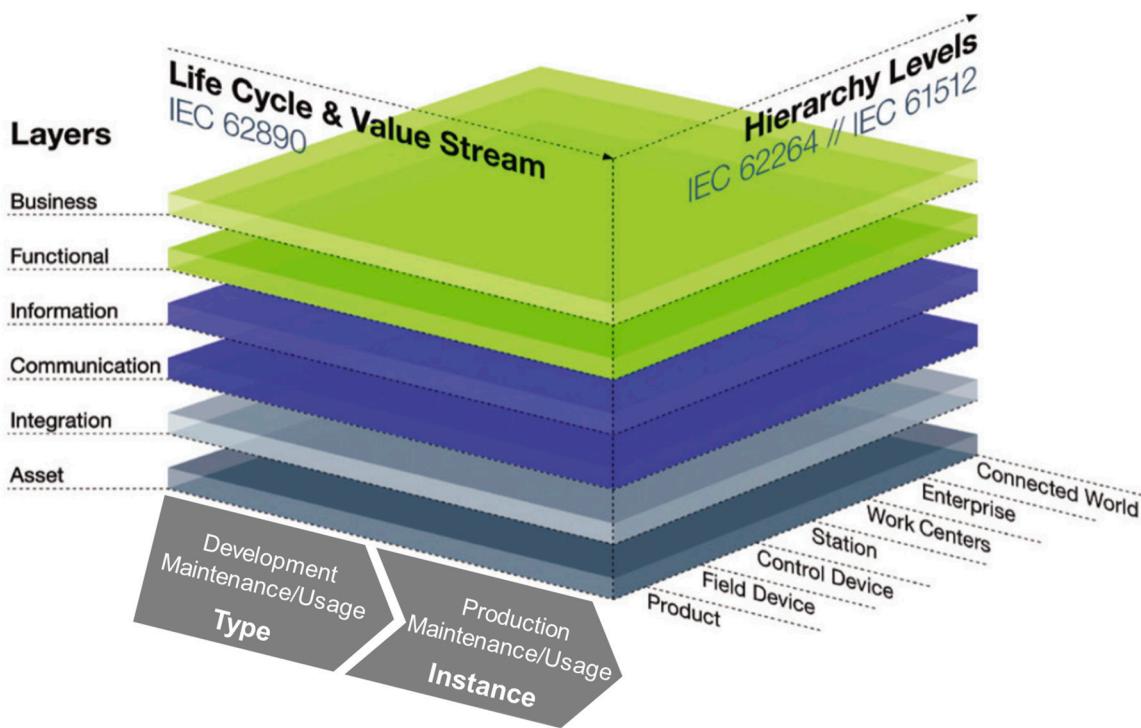


Figure 1. Reference Architectural Model Industry 4.0 (RAMI 4.0) [21].

2.2. The Food and Beverage Industry

Three different types of the production process can be found in the food and beverage industry [23]: batch process for the production of finite quantities of material by subjecting quantities of input materials; continuous process for the continuous material flow through processing equipment; discrete process for the production of units moving between processing stations. The differences of the food and beverage industry from the other manufacturing industries can be summarized as [24–26]: (i) the splitting and mixing lots lead to a combination of divergent processes with convergent processes in the production; (ii) production yields are uncertain as the property of materials is changing over time; (iii) products or semi-finished products are recycled during the production process; (iv) final products have a limited shelf-life and deteriorate over time.

To ensure the quality and safety of the final products, the transparency through the production must be enhanced to achieve efficient process management, which can be supported by the MES, as the MES is connected to the shop floor and serves as the center delivering critical process information to the co-systems [27]. Though researchers have analyzed the factors that can influence the choice of consumers for food and beverage [28], the price remains the most critical factor [29]. Due to the low-profit margins, sectors in the food and beverage industry, for reasonable pricing of the final products, have paid more attention to reduce the energy consumption and enhance the production efficiency [30]. By understanding the consumption of energy resources globally, the MES can contribute to managing and reducing energy consumption in the manufacturing processes [31]. MES can also support the application of lean principles, as it provides real-time information to fees and/or validate the lean processes to improve the efficiency of the production [32]. To satisfy the diverse demands of the consumers, the variety of food and beverage products is growing in the market [33]. By the continuous introduction of new products, the traditional ones can lose their viability in an unpredictable way [34]. The manufacturers have to shift their production strategy from make-to-stock to make-to-order to improve production flexibility, scheduling timeliness, and further to react to the product dynamics. The MES is considered as the basis for scheduling in the process industry, as the information from the whole enterprise, such as the order delivery time, the use of processing materials, and the machine

availability, must be coupled to generate the production plans [35,36]. In the new industrial automation environment, owing to its essential role as information center for data processing and providing between the enterprise level and the automation layer, the modern technologies, such as Industry 4.0, cyber-physical systems, and digital factory, are enabled by the MES to optimize the food and beverage manufacturing processes [37,38].

2.3. MDE for Production Management Systems

To promote the reusability and to reduce the complexity of the software development process, the concept of MDE has been researched and used in a wide range of industrial applications [39], such as business imaging systems, electric systems for telecommunications, and robot operating systems [40,41]. For the development of control systems, Alvarez et al. [42] developed a model-driven approach named MeIA, which has integrated methods and techniques within the automation discipline, provided more accessible verification procedures and structured designs, and improved productivity by means of model reuse and code generation. In the area of business process management, the MDE proposed a set of methodologies, which can bridge the gap between business analysts and software developers [43]. Blal and Leshob [44] proposed a model-driven method to generate and specify services of service-oriented systems from business models expressed by Business Process Model and Notation (BPMN), a standardized graphical specification language for business process modeling and automation [45]. Due to the support of the modular engineering process, management of heterogeneity and complexity, reusability of design artifacts, MDE should not be waived to enable technologies of Industry 4.0 [46]. An MDE framework (MDE4IoT) was introduced in [47] to generate Internet of Things (IoT) systems supporting the modeling of Things and self-adaptation of connected systems. Leal et al. [48] developed a model-driven approach (smartHMI4I4) providing reusable Application Programming Interface (API) and widgets as a global framework to guide the HMI design and generation for different devices across Industry 4.0 application scenarios. Binder et al. [49] proposed a model-driven system development process integrating RAMI 4.0 to provide a common basis for the development of Industry 4.0 systems reducing the complexity of engineering processes and heterogeneity of platforms and toolsets.

For production management systems, some MDE approaches have already been established. For the implementation of an ERP system, a model-driven approach was proposed using the MDA framework by Dugeril and Gaillard [50]. They defined a UML profile to model the business process in the level of ERP and used models to describe the transformation. In their approach, the ERP system was already implemented as a toolbox including different functional components. Instead of code generation, the components can be enabled or disabled to fulfill the requirements of the business process. A modeling approach that uses a state chart for describing and simulating the sequence of operation in MES was presented in [51]. However, the focus was on the modeling and simulation phase, and it lacked further steps for generating an operational system. Oliveira et al. [52] proposed an interpretive MDE approach for enterprise applications and compared the productivity, profitability, and return on investment (ROI) between the generative and interpretive MDE approaches based on its application to development of ERP systems. In this research, though the MDE concept has been implemented, the used modeling language, models and their specifications, and the transformation between them were not clarified. For the MES, an approach according to the MDA development method was presented by Mizuoka and Koga [53], in which the UML and XML Metadata Interchange (XMI) were used for modeling and platforms transforming. The application scenario of this approach was not declared, and it was also mentioned that the UML as a general modeling language may not be suitable for modeling MES [54], because the MES modeling is related to the target business process and requires interdisciplinary information from the co-workers of different departments, who may have different views on the same production processes [55].

The MES Modeling Language (MES-ML) was developed as a modeling language for the interdisciplinary specification of MES [56]. For the generation of operational MES according to

the MDE concept, the metamodel of MES-ML was extended [57]. With this extension, the components of MES (plants, processes, MES functions, and reports) and their connections could be modeled completely, and the extended MES-ML was validated with a modeling example, which proved the feasibility of this language for the MDE of MES. However, it lacked an explanation of the steps that are needed to represent the information in the model and for the transformation between different models for automatic MES generation. The concept of model-driven engineering for MES by using the extended MES-ML was proposed first in 2017 [58]. Chen et al. [59] proposed in 2018 a model-driven approach for MES with more details, which consisted of three steps, modeling, specifying, and generating. Their research focused on the step of modeling with extended MES-ML that was integrated, in which libraries for predefined modeling elements were introduced. The feasibility and the necessity of the predefined modeling elements were also evaluated and proven by their research so that the basis for the model-driven engineering of MES was established. However, details of the latter steps, including specifying and generating, were not clarified.

3. Requirements on the Approach

Conventionally, seven phases are necessary to implement the MES in a manufacturing enterprise [60]: basic determination for determining the MES project scope and content; preplanning for analyzing the business, production processes, and the MES functions; basic planning for evaluating and drafting the MES concept and recommendation; realization planning for mapping the MES in detail; implementation for installing and integrating the hardware and software; commissioning for testing the MES software and training the users; completion for compiling the final report and drafting project accounting. The specific business and manufacturing processes must be considered in each phase to be executed, therefore, the MES must be adopted with considerable customizing and programming effort, which is cost-intensive and error-prone [61,62].

Based on the facts that more than 99% of the enterprises in the food and beverage industry are small- and medium-sized enterprises [63], and few rudiments of MES are specifically designed to fulfill the requirements from the food and beverage industry with relatively low programming and customizing effort, and further low cost, the MES implementation in the food and beverage is hindered by the unavailability of the resources from the side of the end-users and the lack of methods to reduce the implementation complexity from the side of the MES providers. Chen and Voigt [64] analyzed the barriers for the MES implementation in the food and beverage industry combined with their characteristics, and proposed solutions to deal with the barriers, i.e., the use of standardized information model for the information exchange in the whole enterprise to reduce the integrating effort, the service-oriented architecture for dividing the MES functions in independent services to achieve higher reusability, and the model-driven concept for engineering the MES to reduce the programming and customizing effort. Therefore, the requirements that should be fulfilled by the presented approach are defined as:

- Requirement 1 (R1): Development of a feasible model-driven approach for the MES. The model-driven approach should be designed so that it contains the necessary phases to carry the model information from graphical modeling of the desired MES to specifying the solution details and then to the automatic generation of MES functions as well as their application according to the target business processes. Furthermore, the contents of the used platforms in each phase and their transformations must be declared to ensure the interoperability of each phase and the correctness of the generated MES.
- Requirement 2 (R2): Representation of relevant MES functions in the food and beverage industry. The manufacturers in the food and beverage industry need support from the MES to face the challenges of reducing energy consumption, increasing production efficiency, improving production flexibility, and remaining competitive. Therefore, the model-driven approach should be able to represent the MES functions required by the manufacturing enterprises in the food and beverage industry, such as energy management, KPIs calculation, production scheduling, and predictive maintenance. Furthermore, considering the limited available resources of the food

and beverage manufacturing enterprises, the approach should be able to lower the programming and customizing efforts during the engineering process.

- Requirement 3 (R3): Capability of data flow. To implement the developed model-driven approach in different application scenarios without redefining the information interface, a consistent communication standard between the shop floor level and MES level should be supported. Additionally, a standardized communication interface should be used for data exchange to ensure the portability of the modeling elements, which are predefined and to be reused.
- Requirement 4 (R4): Automatic adoption of the generated MES. The model-driven approach should not be defined to serve definite application scenarios but could be applied to fulfill various demands on the MES from different domains. Thus, the basic elements that were used to establish the MES model must be defined neutrally. Due to the flexible sequence of the basic elements and further the information flow, the MES that modeled with the basic elements should be generated dynamically, so that it can be adopted automatically to each specific application scenario.

4. Approach for the Model-Driven Engineering of MES

This section presents each phase of the model-driven approach in detail, i.e., the analyzing phase, the modeling phase, the specifying phase, the generating phase, and the application phase (Figure 2).

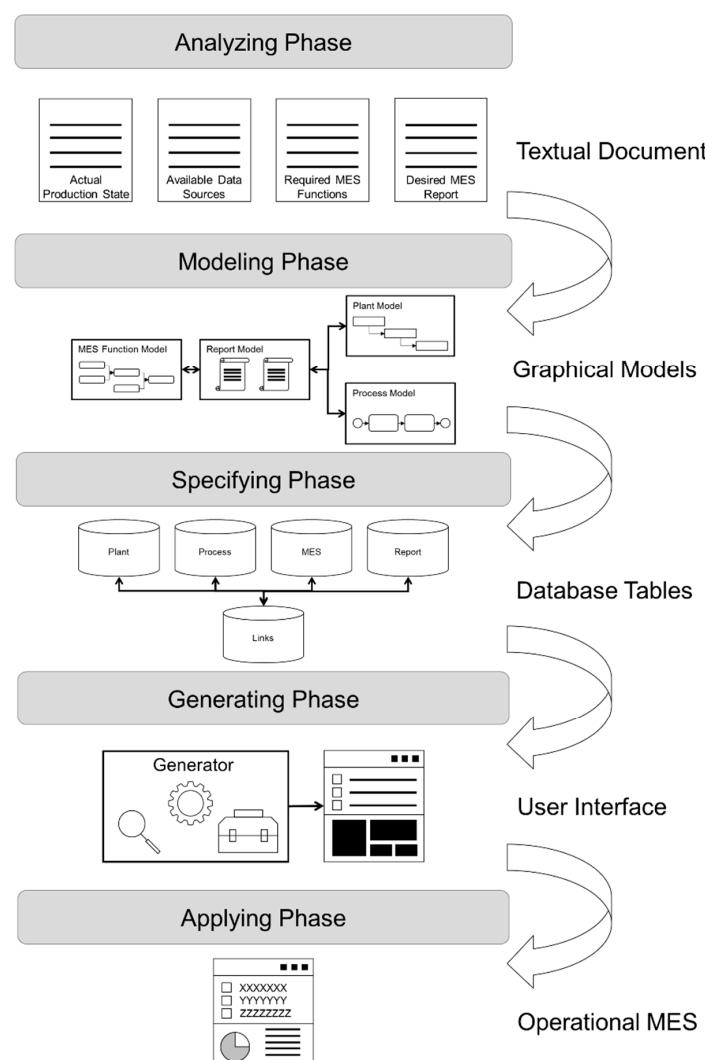


Figure 2. Five phases of the approach for model-driven engineering of Manufacturing Execution Systems (MES).

To ensure the consistency of data flow in the horizontal transformation from model to the operable system, as well as in the vertical integration between the automation level and enterprise level, “Weihenstephan Standards (WS)” [65] are integrated in this approach, which are widely used in the food and beverage industry. The WS is a standard information model that defines a universal communication interface for the communication between different process control systems and the MES. With WS, the processing of the machine and process data could be consistently defined for the MES functions. To ensure the portability of models, WS is used as the primary information model through the whole approach. The following sections present the engineering processes, the tasks, and the related tools of each phase.

4.1. Analyzing Phase

As the preliminary study before the modeling phase, the actual state of the production plant and its processes with their acquirable data sources, the demands on the MES functions, and the desired MES reports should be analyzed. This phase is addressed primarily at the end-users. The main concepts and entities of the domain are analyzed. Multiple co-workers (end-users with internal and/or external solution analysts) with different focuses and working backgrounds must work together to figure out the proper MES functions as well as the information flow among departments in a completed MES [56]. The results of this phase are textual specification sheets indicating the actual stand of the production state (technical systems and production processes), the available data sources from the production, the required functions, and the desired report of the MES.

4.2. Modeling Phase

At the modeling phase, the MES that fulfill the demands in the first phase must be modeled, based on the results of the analyzing phase. The extended MES Modeling Language (MES-ML) is a formal modeling language and was proven to be suitable for modeling the MES with a model-driven concept [57]. An editor that supports the extended MES-ML can assist the modeling process. The output of the modeling phase is a completed graphical model of the MES containing the necessary information for the transformation and generation in the later phases.

In the extended MES-ML, the plant model is structured as a tree diagram with six hierarchy levels to present the technical systems that are inspired by the ISA-95 [15,57], i.e., factory, area, plant, line, machine, and aggregate. Under each technical element in the hierarchy level, data points can be assigned to indicate the data that can be provided. The process can be modeled with three hierarchy levels with an increasing grade of detail; namely, process, process stage, and process operation, which are defined in ISA-88 as the first three levels in the process model [23]. To model the MES function, a series of basic functions were defined to compose the final MES function. The report is modeled with its name and the report elements belong to it. The data from the technical systems and the processes are the inputs, and the results of the MES functions are outputs of the reports. Each report element can display the information in a certain manner, such as text, table, or diagram, and is connected to an MES function providing the related outputs. Figure 3 presents the metamodel of the extended MES-ML for modeling the four components.

It should be noted that to ensure the independence and generality of each model, the connections between the models are established by the mapping function provided by the report model. Although the necessary data from technical elements and/or processes to be processed by the MES functions are modeled, the sources of them are not fixed in the modeling phase. In other words, for any technical elements and processes, which can provide the related data, the appropriate MES functions can be executed, i.e., the reusability can be ensured by the division of the data from the MES functions. By defining the inputs and outputs of the reports, the connections among the plant model, process model, and the MES function model are firstly established.

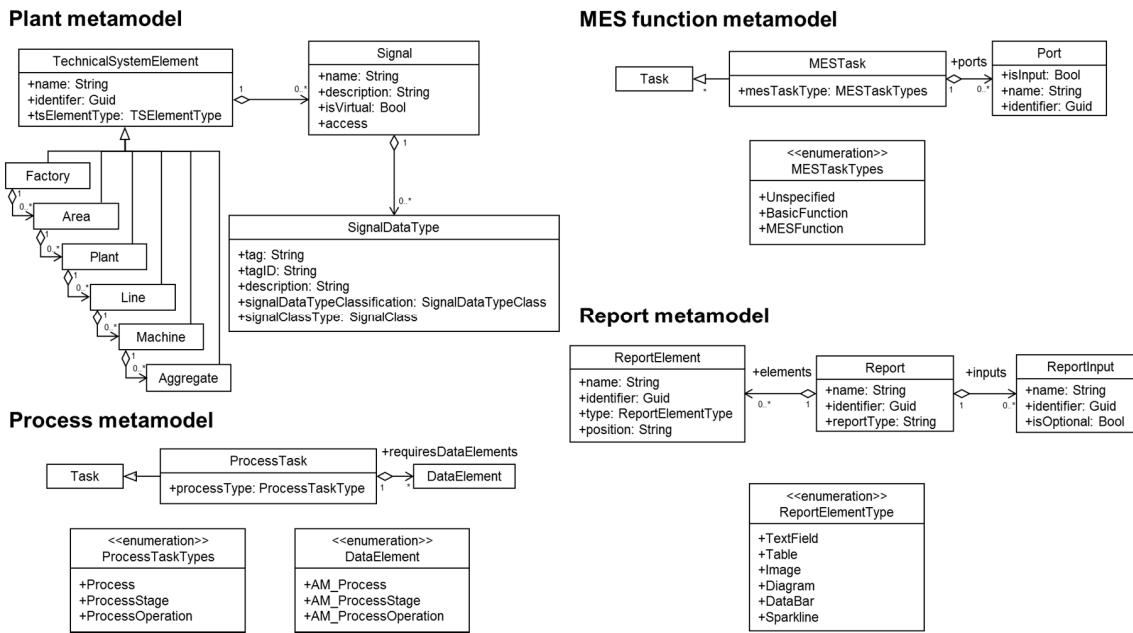


Figure 3. Metamodel of the modeling language MES Modeling Language (MES-ML) [57,59].

4.3. Specifying Phase

Following the modeling phase, taking the graphical MES model as the input, an MES specification as the output of the specifying phase is filled. The primary task of this phase is to transform the information from the graphical models into a format that software can utilize. The database tables are the platform for the MES specification. Different tables related to the four models in the modeling phase are defined to represent the information in the graphical model. In this phase, the transformation process is a mapping process, as the structure, content, and the relationship of the tables were defined as a one-to-one mapping of the extended MES-ML metamodel. This phase can be performed automatically by using a mapping tool that knows the relationship between the metamodel of the extended MES-ML and the database tables.

In the following, the related logical entity-relationship diagrams (ERD) of the database are presented using Crow's Foot Notation. As the technical systems are modeled using different hierarchies and the data points for collection can be assigned to each element in the plant model (according to the meta model of MES-ML), two tables named “Location” and “LocationDataPoint” are used to describe the plant model (Figure 4, left). The table “Location” contains attributes that are identical to the hierarchy level of the plant metamodel with factory, area, plant, line, machine, and aggregate being used to present the modeling elements on each hierarchy. The table “LocationDataPoint” presents the data points on each element. The types and descriptions of the data points can also be found with the attributes “DataPointType” and “DataPointDescription”. Besides that, the attribute named “LocationLink” is made to indicate the relationship between a specific data point and its host element. Similar to the plant model, the information in the process model can be represented with two tables named “Process” and “ProcessDataPoint” (Figure 4, right). The name of the modeling elements on each hierarchy level in the process model can be found in the “Process” table. The type of the used process data points and their link to the process element can be described with the attributes “DataPointType” and “ProcessLink.” Another table to indicate the correlation between the plant model and the process model is also defined (Figure 4, middle). By using the two attributes “ProcessLink” and “LocationLink” in the “LocationProcess” table, the link between the processes and technical systems can be found.

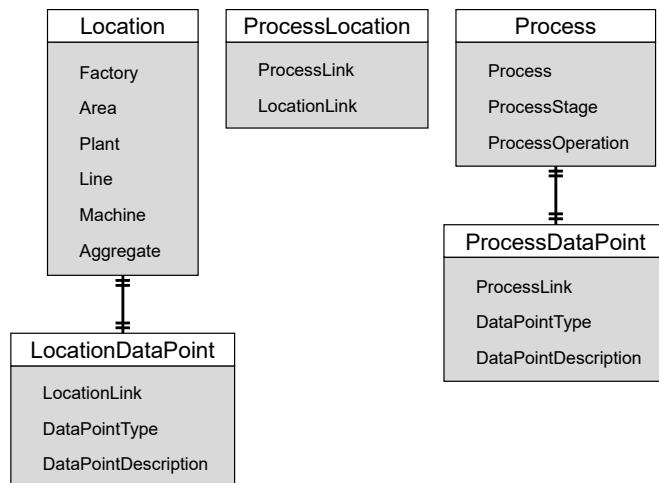


Figure 4. Entity-relationship diagrams (ERD) of tables for plant model, process model, and the link between them.

As the MES functions are composed of the basic functions, there are six tables to represent the information from the basic functions, MES functions, and the logical relationship between them. On the right side of Figure 5, the basic functions used to compose the MES functions and their input parameters are described in the tables named “BasicFunction” and “BasicFunctionInput.” On the left side are the tables named “MESFunction” and “MESFunctionParameter” to represent the name of the MES function and the necessary attributes of the total input and output parameters to realize this MES function, such as the “Direction” to differentiate the input (Direction = 1) and output (Direction = 0) parameters. The two tables in the middle, namely the “MESFunctionInstance” and “MESFunctionInstanceParameter,” serve as bridges in the specification for the MES function model to indicate on one side, the affiliation between the basic functions and the MES functions and on the other side, the assignment of the input and output parameters of the MES functions to the input parameters of the basic functions.

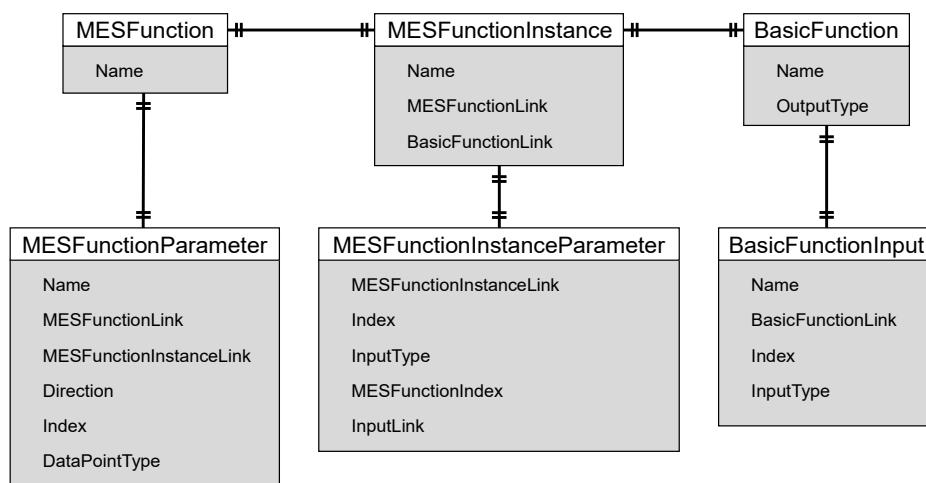


Figure 5. ERD of tables for MES function model.

The information in the report model can be transformed using the four tables that are shown in Figure 6. The table “Report” contains the name and the type of report. As a report can contain several report elements to present the results of different MES functions, and each MES function needs different data sources from the plant model and/or process model, the tables “ReportElement” and “ReportInput” are defined. The table “ReportElementOutput” is responsible for establishing the link between the report elements and the MES functions.

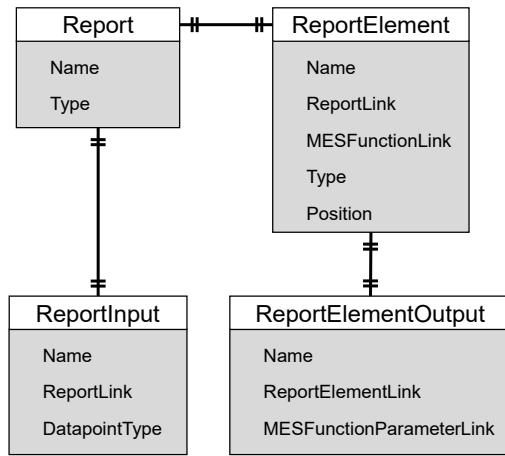


Figure 6. ERD of tables for report model.

4.4. Generating Phase

In this phase, based on the MES specification, MES with the demanded functions (without specific values for input parameters) is generated automatically with the help of a generator. The generator has two components, the front-end and the back-end. The front-end is a graphical user interface for the end-user to parameterize the inputs of the MES. The back-end, which realizes the data processing of the MES, can be divided into two parts, i.e., a toolbox and a connection finder.

The design of the graphical user interface is dynamically dependent on the MES specification. It contains two areas; input area and output area. In the input area, the input parameters are listed, and the values of input parameters are ready to be chosen and/or modified by the end-user. After the MES functions have processed the data according to the given values of input parameters, the results are assigned as the values of output parameters, shown in the output area (Figure 7).

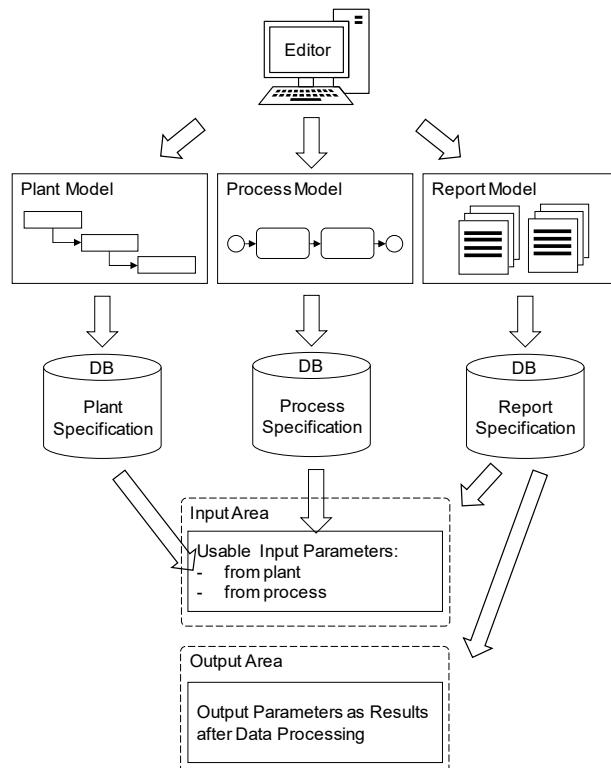


Figure 7. The front-end and its correspondence with the specification.

For the model-driven engineering of the software system that must be individualized, the MDA framework resembles the best practices: the first design of the business process is to be implemented and then processed with parameterization [50]. In this sense, the necessary MES functions are already implemented in the generating phase, while the specific sources of data (input parameters) that must proceed are not defined. This concept was proposed for the customization of ERP using a model-driven approach: there is no component to be generated or removed, but to be enabled/disabled. What needs to be generated is the constraints information for the parameterization to configure the system [50]. In the modeling and specification phases, no real function is implemented, but a description for the inputs and outputs of the functions, as well as the name of the basic functions and their connections with each other. The basic functions for data processing are defined in [59] and were firstly implemented as executable procedures in the toolbox. The connection finder that can invoke the basic functions from the toolbox is responsible for reproducing the sequence flow of basic functions to ensure the correct order of value passing among the basic functions (Figure 8).

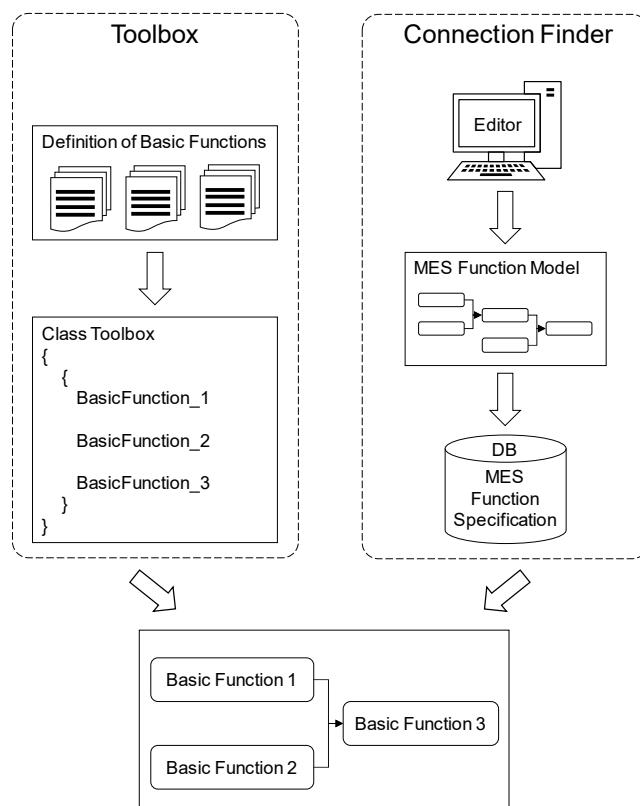


Figure 8. The back-end of the generator for data processing.

4.5. Application Phase

In the application phase, according to real business processes, the demands on the MES and the inputs are parameterized by the end-user so that the desired specific MES can be created. Generally, the data processing can be divided into four steps. First, retrieving the needed basic functions from the toolbox; secondly, assigning the input parameters from the input area to the basic functions; thirdly, processing the input parameters using the modeled sequence of basic functions and fourthly, showing the results as parameters in the output area of the graphical user interface.

5. Use Case: Energy Management for the Beer Brewing Process

As mentioned in Section 2.1, the most demanded MES functions are the functions that can support the improvement of the production efficiency, tracking and tracing, and energy consumption [66].

Chen et al. [59] also defined the modeling elements to manage energy consumption and analyze production efficiency that is relevant to the food and beverage industry. Based on their results, to evaluate the feasibility of the presented approach and the achievement of the requirements defined in Section 3, the approach was applied to the traditional brewing process in a fictitious brewhouse. This brewhouse provides the basis of the technical systems and production process that the generated MES is built for, and data for processing. The brewing process was chosen as it is a representative production type (batch process) in the food and beverage industry. Figure 9 illustrates the traditional brewing process in the brewhouse.

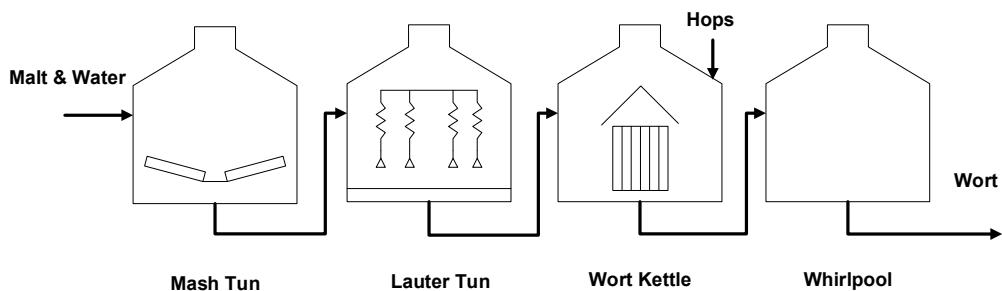


Figure 9. Traditional brewing process in the brewhouse.

In the brewhouse, to produce the wort for fermentation, malted barley, water, and hops are needed as raw materials. The malt and water are mixed and heated to specific temperature for extracting in the mash tun. The separation of the liquid phase (wort) from the grains is performed in the lauter tun. The wort kettle boils the wort together with hops for their sterilization and isomerization. Solid particles in wort, such as hops and proteins, are precipitated and separated at the bottom of the whirlpool. After that, the wort is cooled down and is ready for fermentation and storage in the fermentation room [67].

5.1. Analyzing the Demands and Actual State

As the brewhouse is the main energy consumer of the whole brewery, namely 38% of total energy consumption [68,69], and the energy used in a brewhouse can be divided into two primary units, thermal energy, and electrical energy, each for 10.2–11.4 kWh/hl (hectoliter) SB (sales beer) and 0.84–2.3 kWh/hl SB in average [70,71], the targeted MES should focus on the management of the thermal energy in this use case. The wort kettle and the mash tun are the consumers of the thermal energy in the brewhouse, taking the share of 77.1% and 22.9% [70]. According to the data above, the consumption data of the brewery with a designed annual beer production volume of 200,000 hl are shown in Table 1. The consumption data with the data point defined in WS (WS_Cons_Steam) were stored in a database for the period from 1 January 2019 00:00:00 to 31 December 2019 23:59:59.

Table 1. Thermal energy consumption data of the consumers in the brewhouse.

Energy Consumer	Thermal Energy Consumption from the Literature on Average	Consumption per Year	Consumption per Second
Mash Tun	2.48 kWh/hl SB	496 MWh	0.02 kWh
Wort Kettle	8.32 kWh/hl SB	1664 MWh	0.07 kWh

5.2. Modeling the MES

The components in this use case were modeled in an editor that supports the extended MES-ML presented in [57,58]. The modeling elements that were defined in [59] were used. The production line in the brewery includes the mash tun, lauter tun, wort kettle, whirlpool, and heat exchanger. As mentioned above, the related data points for the energy data were assigned to the technical systems

in the plant model. Figure 10 shows the model of the technical systems with a detailed view of the data point assigned to the mash tun.

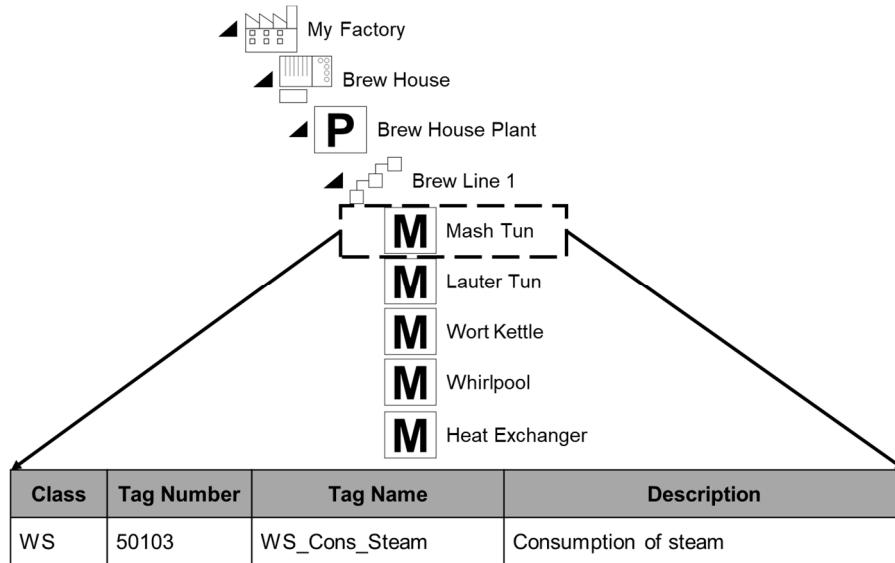


Figure 10. Plant model of the brewhouse.

The overall process of wort production in the brewery was described at the hierarchy level of process. This process was presented in greater detail at the process stage hierarchy level. Each element in the process stage was modeled with more details at the process operation hierarchy level. Figure 11 shows the resulted process model.

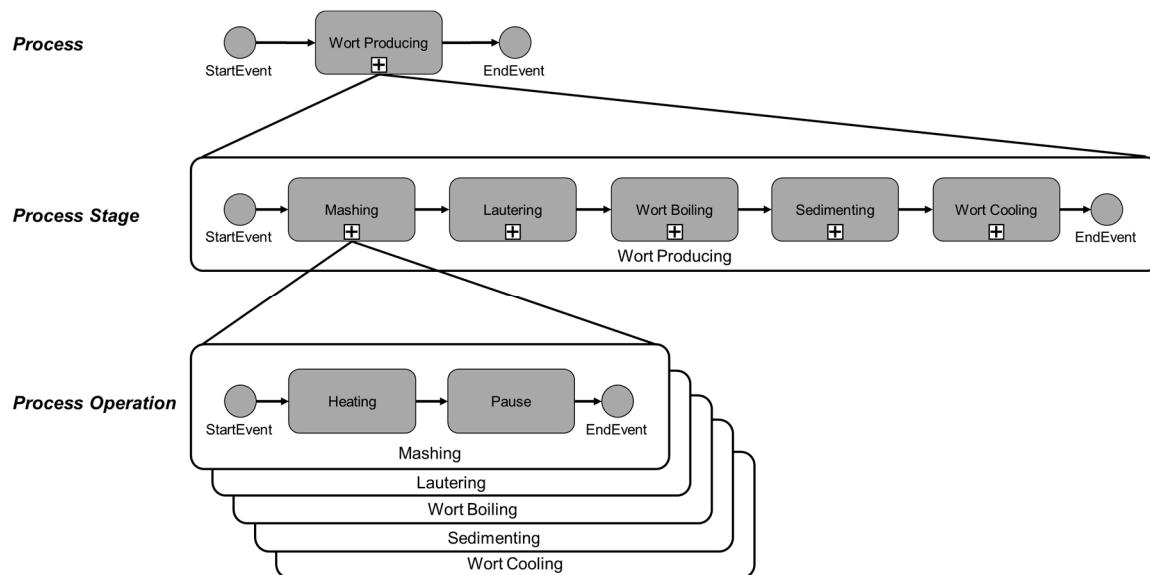


Figure 11. Process model of the brewing process.

In this use case, two MES functions were modeled: energy consumption of a specified machine in a given period of time, and the sum of the energy consumption of two machines in a given period of time.

Figure 12 shows the model for the second MES function, which was made up of two basic functions named “EnergyConsumptionCalculation” to calculate the energy consumption of a machine in a given

period of time, and “ $\text{Sum}(x_1, x_2)$ ” to sum the two consumption values. For the automatic generation of the desired MES functions, these two basic functions should be implemented in the toolbox of the generator.

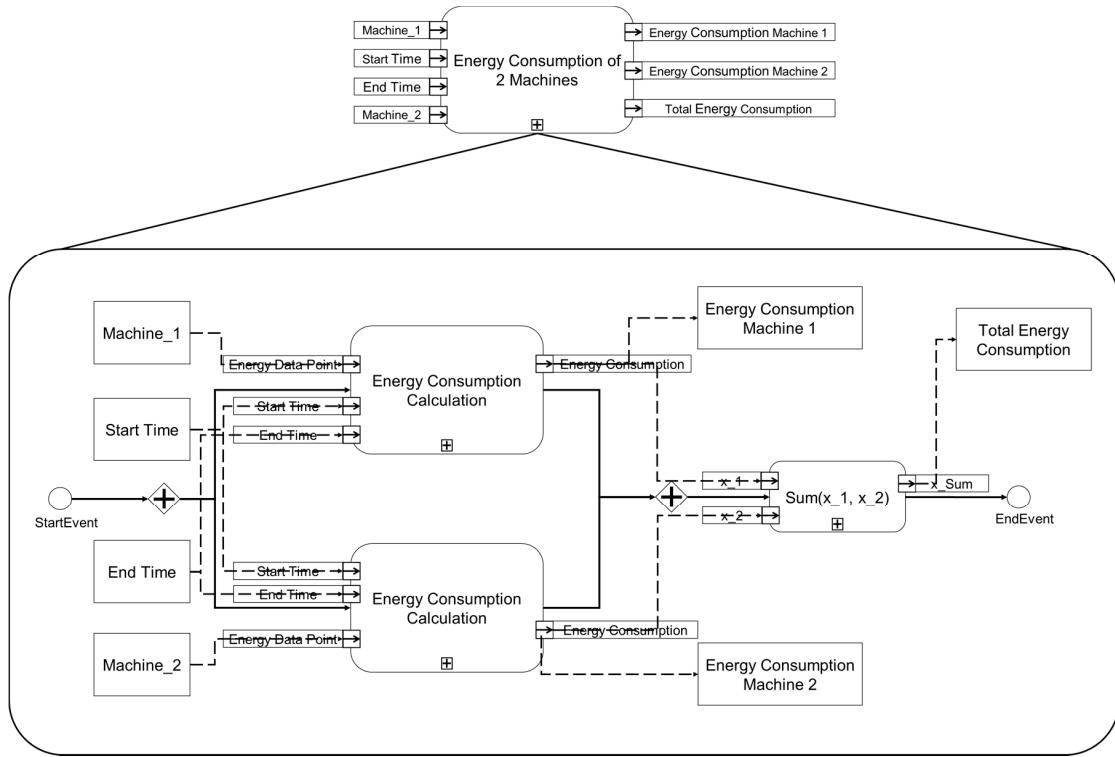


Figure 12. MES function model to fulfill the demands on the MES for energy management.

Figure 13 shows the related report element for the MES function “Energy Consumption of 2 Machines.” The results of the MES function, namely the energy consumption of machine 1, the energy consumption of machine 2, and the total energy consumption, were considered as the outputs of the final report in text form.

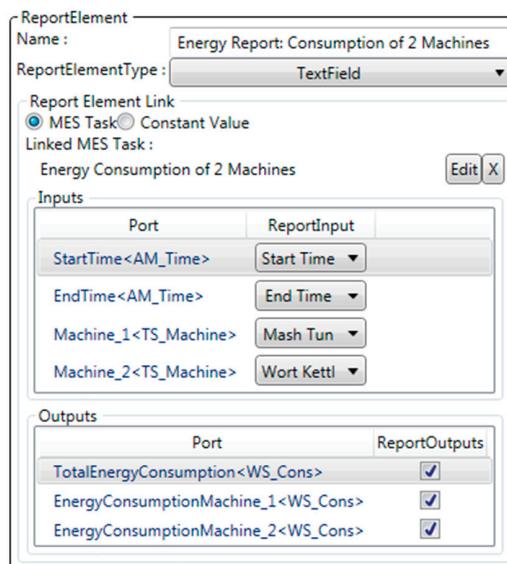


Figure 13. Element in the report model related to the MES function for calculating the energy consumption.

5.3. Specifying the Information from the Graphical Model

After the components of the MES were modeled, they were transformed into the database tables of the specification by the mapping function of the editor used in Section 5.2. As an example, to clarify the data structure of the specification, Tables 2 and 3 present the specification for the used basic functions and the description of their input parameters in the database tables named “BasicFunction” and “BasicFunctionInput”.

Table 2. Specification in the “BasicFunction” table.

Key	Name	OutputType
1	EnergyConsumptionCalculation	Consumption
2	Sum(x_1, x_2)	Sum

Table 3. Specification in the “BasicFunctionInput” table.

Key	Name	BasicFunctionLink	Index	InputType
1	StartTime	1	0	StartTime
2	EndTime	1	1	EndTime
3	EnergyDataPoint	1	2	EnergyData
4	x_1	2	0	Value
5	x_2	2	1	Value

From the two tables we can reproduce the information. To compose the MES functions in this MES, two basic functions are needed, namely the “EnergyConsumptionCalculation” and “Sum(x_1, x_2)”; the first basic function (“Key” = 1 in Table 2) has three input parameters named “StartTime,” “EndTime,” and “EnergyDataPoint” (“BasicFunctionLink” = 1 in Table 3); the second basic function (“Key” = 2 in Table 2) has two input parameters named “ x_1 ” and “ x_2 ” (“BasicFunctionLink” = 2 in Table 3).

5.4. Generating the MES

Visual C# in Microsoft Visual Studio was chosen as the platform for the generator in this use case. For data processing, the implemented basic functions have access to the energy consumption database. Figure 14 presents the user interface of the generator at the beginning (left) and the report to be parameterized that was generated based on the specification (right). The input area contained text boxes to modify the values of input parameters. The output area was designed to indicate the energy consumption of each machine and their total consumption. To generate the user interface with input and output area, and to insert the parameters/components in each area, the information in tables named “MESFunction” and “MESFunctionParameter” should be utilized by the front-end of the generator.

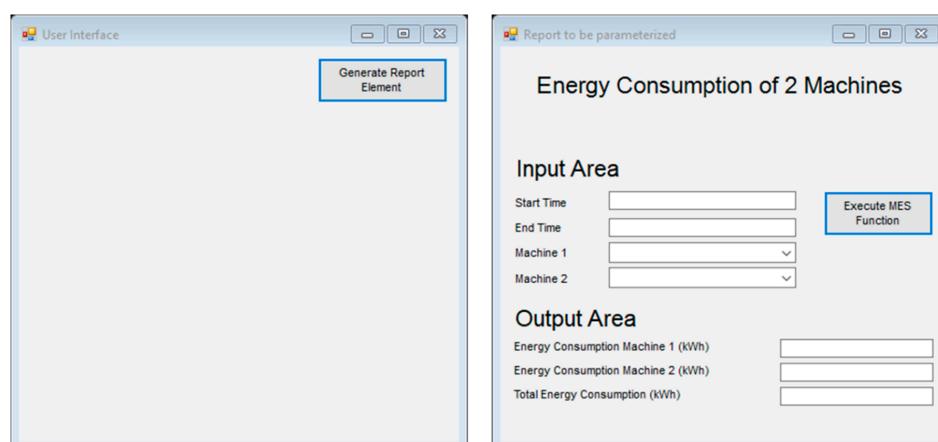


Figure 14. User interface of the generator (**left**) and the generated report to be parameterized (**right**).

5.5. Application of the MES

At the application phase, the input parameters were modified by the end-users according to their specific demands on the MES. As the mash tun and the wort kettle are the main thermal energy consumers, the input parameters of the generated MES were modified to calculate the thermal energy consumption of the two machines for a given period of time. Figure 15 illustrates the modified input area (left) and the results after data processing in the output area (right) pertaining to these two machines. The information in tables “BasicFunction”, “BasicFunctionInput”, “MESFunctionInstance”, “MESFunctionParameter”, and “MESFunctionInstanceParameter” was read by the back-end of the generator to invoke the procedures of the basic functions and pass the values between them.

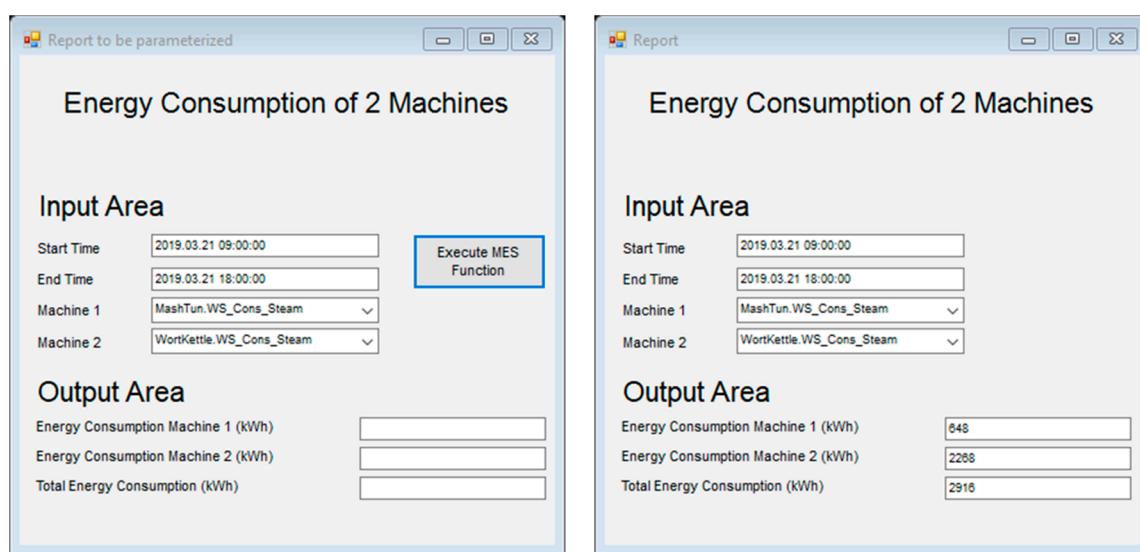


Figure 15. MES report with modified input area (**left**) and results in the output area (**right**).

5.6. Evaluation

In this use case on the brewing process in a brewhouse, after the requirements were analyzed, the MES for the management of the energy consumption was modeled using the modeling language, the extended MES-ML. This graphical model was transformed automatically into a specification in a series of predefined database tables. The information in the original model was reproduced by the specification resulting in the basis that was utilized by the MES generator. On one hand, the generator created a user interface based on the plant model, process model, and report model to establish communication between the end-user and the software. On the other hand, the internal connections of the basic functions for data processing were created with the help of the toolbox and the connection finder of the generator. As the possible input parameters/data sources can be chosen in the user interface according to the real business process, the generated solution can be parameterized specifically. Results according to the specific requirements of the business process were shown in the output area of the user interface. The mash tun and wort kettle each respectively consume 0.07 kWh/s and 0.02 kWh/s of the thermal energy in the brewhouse, which in return theoretically causes the consumption of 2268 kWh and 648 kWh in nine hours. The results calculated by the generated MES were identical to the theoretically expected values.

Based on the use case described above, and according to the requirements defined at the beginning of this paper, the presented approach can be evaluated. The model-driven approach for the engineering of the MES that was presented in Section 4 has been proven feasible by this use case, which clarifies each phase in this approach and the transformation between them (R1). The MES with the function to manage the thermal energy consumption in the brewhouse has been generated in the use case. Though the energy management is a relevant MES function for the food and beverage industry,

to evaluate if other relevant MES functions in the food and beverage industry can be generated with this approach, the continuous process and the discrete process should be considered on one hand, as the use case focused on the batch process and the requirements, although the MES functions can be the same, the data availability, source, and processing may vary with the process type; on the other hand, more modeling elements, basic functions, and MES functions should be defined, applied, and validated with this approach. Once the solution of the MES was finished to be modeled in the modeling phase, the transformations from a graphical model to a software readable specification and the generation of an MES can be made without any other manual programming effort (R2). The WS were integrated into this approach to ensure data consistency. This offers standardized data signals for a consistent communication between the machines and MES, and ensures machine- and process-independent modeling of the MES functions as well as standardized data processing. The WS are the information basis for the whole model-driven approach, also for the further horizontal and vertical communication in the future (R3). Additionally, we have tested different application scenarios with the presented approach. It was confirmed that the MES with functions, which can be composed of the basic functions that are already implemented in the toolbox of the MES generator, can be generated automatically. In this sense, further adaption and expansion of the generated MES can be performed in the modeling phase, which reduces the customization effort, as an adopted and/or extended MES can then be generated automatically again (R4). Although two prototypes of software tools were used in the use case, i.e., the editor with mapping function to assist the modeling process and perform the mapping process, and the generator to generate the user interface and process the energy data, it must be indicated that there can be different variants of technology to realize the functionality of the used software tools, as the metamodel of the modeling language, the mapping from model to specification, and the methodology of the MES generation have been clarified in detail in Section 4.

6. Conclusions and Outlook

This paper presented an approach for the automatic generation of MES according to the model-driven concept with low programming and customizing effort. The presented approach has filled the theoretical gap in the research area for integrating the model-driven concept for the development of the MES, and further for other IT systems in the manufacturing enterprises. With the focus of its application, it can benefit the manufacturers in the food and beverage industry with limited flexibility to invest in conventional MES projects. The use case with the automatic generation of the MES for thermal energy management in the brewhouse has proven the feasibility and practicability of the presented approach.

Although the requirements were evaluated as fulfilled by the presented approach, as the first rudiment for the model-driven MES engineering in the food and beverage industry, there are still some limitations: (i) the use case focused on the MES function for energy management based on historical acquired data, the functions that concern the “execution” side (real-time reaction) of the MES were not considered, which can also be relevant for the food and beverage industry, such as the detailed scheduling, process management, and product tracking; (ii) the approach has not been validated with real production data, which may deliver meaningful potential areas for the improvement of the presented approach; (iii) although the standardized information model has been applied for consistent data communication, the integration of the MES with shop floor control systems and management systems on the enterprise level has not been considered in the developed approach. From the viewpoint of the software development, the prototype of the generator should be expanded and tested regarding to the computational complexity.

In future work, libraries should be enriched with more modeling elements for different application domains in the food and beverage industry to verify the interoperability of the presented approach. Additionally, the range of the predefined basic functions must be extended so that more MES functions with different focuses can be modeled and generated. Furthermore, MES functions for the execution of the process, such as dispatching production, production tracking, and detailed scheduling, should be

applied with real production data for a more convincing validation of the whole approach. It is also planned to define the interface and key data for the communication between the MES, the control systems, and the enterprise management systems.

Author Contributions: Conceptualization, X.C.; Methodology, X.C. and C.N.; Validation, X.C.; Writing—Original Draft, X.C.; Writing—Review & Editing X.C., C.N. and T.V.; Supervision, T.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Wang, C.; Chen, X.; Soliman, A.-H.A.; Zhu, Z. RFID Based Manufacturing Process of Cloud MES. *Future Internet* **2018**, *10*, 104. [[CrossRef](#)]
- Tarhini, A.; Ammar, H.; Tarhini, T.; Masa’deh, R. Analysis of the critical success factors for enterprise resource planning implementation from stakeholders’ perspective: A systematic review. *Int. Bus. Res.* **2015**, *8*, 25–40. [[CrossRef](#)]
- Agostinho, C.; Ducq, Y.; Zacharewicz, G.; Sarraipa, J.; Lampathaki, F.; Poler, R.; Jardim-Goncalves, R. Towards a sustainable interoperability in networked enterprise information systems: Trends of knowledge and model-driven technology. *Comput. Ind.* **2016**, *79*, 64–76. [[CrossRef](#)]
- Ramos, A.L.; Ferreira, J.V.; Barceló, J. Model-based systems engineering: An emerging approach for modern systems. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2011**, *42*, 101–111. [[CrossRef](#)]
- Franzago, M.; Di Ruscio, D.; Malavolta, I.; Muccini, H. Collaborative model-driven software engineering: A classification framework and a research map. *IEEE Trans. Softw. Eng.* **2017**, *44*, 1146–1175. [[CrossRef](#)]
- Bézivin, J. On the unification power of models. *Softw. Syst. Modeling* **2005**, *4*, 171–188. [[CrossRef](#)]
- Vyatkin, V. Software engineering in industrial automation: State-of-the-art review. *IEEE Trans. Ind. Inform.* **2013**, *9*, 1234–1249. [[CrossRef](#)]
- Das, N.; Ganesan, S.; Jweda, L.; Bagherzadeh, M.; Hili, N.; Dingel, J. Supporting the model-driven development of real-time embedded systems with run-time monitoring and animation via highly customizable code generation. In Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, Saint-Malo, France, 2–7 October 2016.
- Balagtas-Fernandez, F.T.; Hussmann, H. Model-driven development of mobile applications. In Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering, L’Aquila, Italy, 15–19 September 2008.
- Vogel-Heuser, B.; Fay, A.; Schaefer, I.; Tichy, M. Evolution of software in automated production systems: Challenges and research directions. *J. Syst. Softw.* **2015**, *110*, 54–84. [[CrossRef](#)]
- Jacobs, F.R. Enterprise resource planning (ERP)—A brief history. *J. Oper. Manag.* **2007**, *25*, 357–363. [[CrossRef](#)]
- Waschull, S.; Wortmann, J.C.; Bokhorst, J.A.C. Manufacturing Execution Systems: The Next Level of Automated Control or of Shop-Floor Support? In Proceedings of the IFIP International Conference on Advances in Production Management Systems, Seoul, Korea, 26–30 August 2018.
- MESA. MESA Explained: A High Level Vision. *MESA Int. White Paper 6* **1997**, *1*, 997.
- International Electrotechnical Commission. IEC 62264-1: Models and Terminology; International Electrotechnical Commission: Geneva, Switzerland, 2013.
- International Society of Automation. ANSI/ISA-95.00.01-2000 Enterprise-Control System Integration Part 1: Models and Terminology; International Society of Automation: Research Triangle Park, NC, USA, 2000.
- Bratukhin, A.; Sauter, T. Functional analysis of manufacturing execution system distribution. *IEEE Trans. Ind. Inform.* **2011**, *7*, 740–749. [[CrossRef](#)]
- Xu, L.D.; Xu, E.L.; Li, L. Industry 4.0: State of the art and future trends. *Int. J. Prod. Res.* **2018**, *56*, 2941–2962. [[CrossRef](#)]
- Stock, T.; Seliger, G. Opportunities of sustainable manufacturing in industry 4.0. *Procedia CIRP* **2016**, *40*, 536–541. [[CrossRef](#)]
- Hankel, M. *Industrie 4.0: Das Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0)*; Zentralverband Elektrotechnik- und Elektronikindustrie: Hannover, Germany, 2015.

20. Kannan, S.M.; Suri, K.; Cadavid, J.; Barosan, I.; van den Brand, M.; Alferez, M.; Gerard, S. Towards industry 4.0: Gap analysis between current automotive MES and industry standards using model-based requirement engineering. In Proceedings of the 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Gothenburg, Sweden, 5–7 April 2017.
21. BITKOM; VDMA; ZVEI. *Umsetzungsstrategie Industrie 4.0—Ergebnisbericht der Plattform Industrie 4.0*; BITKOM: Berlin, Germany, 2015.
22. Arica, E.; Powell, D.J. Status and future of manufacturing execution systems. In Proceedings of the 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Singapore, 10–13 December 2017.
23. International Society of Automation. *ANSI/ISA-88.01-2010 Batch Control Part 1: Models and Terminology*; International Society of Automation: Pittsburgh, PA, USA, 2010.
24. Hvolby, H.-H.; Trienekens, J.H. Manufacturing control opportunities in food processing and discrete manufacturing industries. *Int. J. Ind. Eng. Theory Appl.* **1999**, *6*, 6–14.
25. Den Ouden, M.; Dijkhuizen, A.A.; Huirne, R.B.M.; Zuurbier, P.J.P. Vertical cooperation in agricultural production-marketing chains, with special reference to product differentiation in pork. *Agribus. Int. J.* **1996**, *12*, 277–290. [[CrossRef](#)]
26. Trienekens, J.H. Management of Processes in Chains: A Research Framework. Ph.D. Thesis, Wageningen University, Wageningen, Holland, 1999.
27. Moe, T. Perspectives on traceability in food manufacture. *Trends Food Sci. Technol.* **1998**, *9*, 211–214. [[CrossRef](#)]
28. European Food Information Council. *Determinants of Food Choice: EUFIC Review*; EUFIC: Brussels, Belgium, 2005.
29. DiSantis, K.I.; Grier, S.A.; Odoms-Young, A.; Baskin, M.L.; Carter-Edwards, L.; Young, D.R.; Lassiter, V.; Kumanyika, S.K. What “price” means when buying food: Insights from a multisite qualitative study with Black Americans. *Am. J. Public Health* **2013**, *103*, 516–522. [[CrossRef](#)]
30. Olsmats, C.; Kaivo-Oja, J. European packaging industry foresight study—Identifying global drivers and driven packaging industry implications of the global megatrends. *Eur. J. Futures Res.* **2014**, *2*, 39. [[CrossRef](#)]
31. Bunse, K.; Vodicka, M.; Schönsleben, P.; Brülhart, M.; Ernst, F.O. Integrating energy efficiency performance in production management—gap analysis between industrial needs and scientific literature. *J. Clean. Prod.* **2011**, *19*, 667–679. [[CrossRef](#)]
32. Cottyn, J.; van Landeghem, H.; Stockman, K.; Derammelaere, S. A method to align a manufacturing execution system with Lean objectives. *Int. J. Prod. Res.* **2011**, *49*, 4397–4413. [[CrossRef](#)]
33. Matthews, J.; Singh, B.; Mullineux, G.; Medland, T. Constraint-based approach to investigate the process flexibility of food processing equipment. *Comput. Ind. Eng.* **2006**, *51*, 809–820. [[CrossRef](#)]
34. Gargouri, E.; Hammadi, S.; Borne, P. A study of scheduling problem in agro-food manufacturing systems. *Math. Comput. Simul.* **2002**, *60*, 277–291. [[CrossRef](#)]
35. Cupek, R.; Ziebinski, A.; Huczala, L.; Erdogan, H. Agent-based manufacturing execution systems for short-series production scheduling. *Comput. Ind.* **2016**, *82*, 245–258. [[CrossRef](#)]
36. Wauters, T.; Verbeeck, K.; Verstraete, P.; Berghe, G.V.; de Causmaecker, P. Real-world production scheduling for the food industry: An integrated approach. *Eng. Appl. Artif. Intell.* **2012**, *25*, 222–228. [[CrossRef](#)]
37. Vogel-Heuser, B.; Diedrich, C.; Broy, M. Anforderungen an CPS aus Sicht der Automatisierungstechnik. *Automatisierungstechnik* **2013**, *61*, 669–676. [[CrossRef](#)]
38. Vogel-Heuser, B.; Kegel, G.; Wucherer, K. Global information architecture for industrial automation. *Atp Mag.* **2009**, *51*, 108–115. [[CrossRef](#)]
39. Raibulet, C.; Fontana, F.A.; Zanoni, M. Model-driven reverse engineering approaches: A systematic literature review. *IEEE Access* **2017**, *5*, 14516–14542. [[CrossRef](#)]
40. Hutchinson, J.; Rouncefield, M.; Whittle, J. Model-driven engineering practices in industry. In Proceedings of the 33rd International Conference on Software Engineering, Honolulu, HI, USA, 21–28 May 2011.
41. Zander, S.; Heppner, G.; Neugschwandtner, G.; Awad, R.; Essinger, M.; Ahmed, N. A model-driven engineering approach for ros using ontological semantics. *arXiv* **2016**, arXiv:1601.03998.
42. Alvarez, M.L.; Sarachaga, I.; Burgos, A.; Estévez, E.; Marcos, M. A methodological approach to model-driven design and development of automation systems. *IEEE Trans. Autom. Sci. Eng.* **2016**, *15*, 67–79. [[CrossRef](#)]
43. Fabra, J.; de Castro, V.; Álvarez, P.; Marcos, E. Automatic execution of business process models: Exploiting the benefits of model-driven engineering approaches. *J. Syst. Softw.* **2012**, *85*, 607–625. [[CrossRef](#)]

44. Blal, R.; Leshob, A. A model-driven service specification approach from BPMN models. In Proceedings of the 2017 IEEE 14th International Conference on e-Business Engineering (ICEBE), Shanghai, China, 4–6 November 2017.
45. Geiger, M.; Harrer, S.; Lenhard, J.; Wirtz, G. BPMN 2.0: The state of support and implementation. *Future Gener. Comput. Syst.* **2018**, *80*, 250–262. [[CrossRef](#)]
46. Vogel-Heuser, B.; Hess, D. Guest editorial industry 4.0—prerequisites and visions. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 411–413. [[CrossRef](#)]
47. Ciccozzi, F.; Spalazzese, R. MDE4IoT: Supporting the internet of things with model-driven engineering. In Proceedings of the International Symposium on Intelligent and Distributed Computing, Paris, France, 10–12 October 2016.
48. Leal, P.; Madeira, R.N.; Romão, T. Model-Driven Framework for Human Machine Interaction Design in Industry 4.0. In Proceedings of the IFIP Conference on Human-Computer Interaction, Paphos, Cyprus, 2–6 September 2019.
49. Binder, C.; Neureiter, C.; Lastro, G. Towards a model-driven architecture process for developing Industry 4.0 applications. *Int. J. Modeling Optim.* **2019**, *9*, 1–6. [[CrossRef](#)]
50. Dugerdil, P.; Gaillard, G. Model-Driven ERP Implementation. In Proceedings of the 2nd International Workshop on Model-Driven Enterprise Information Systems, Paphos, Cyprus, 23–24 May 2006.
51. Rabbani, M.J.; Ahmad, F.M.; Baladi, J.; Khan, Y.A.; Naqvi, R.A. Modeling and simulation approach for an industrial manufacturing execution system. In Proceedings of the 2013 IEEE 3rd International Conference on System Engineering and Technology, Shah Alam, Malaysia, 19–20 August 2013.
52. Oliveira, A.; Bischoff, V.; Gonçales, L.J.; Farias, K.; Segalotto, M. BRCode: An interpretive model-driven engineering approach for enterprise applications. *Comput. Ind.* **2018**, *96*, 86–97. [[CrossRef](#)]
53. Mizuoka, K.; Koga, M. MDA development of Manufacturing Execution System based on automatic code generation. In Proceedings of the SICE Annual Conference, Taipei, Taiwan, 18–21 August 2010.
54. Ricken, M.; Vogel-Heuser, B. Modeling of manufacturing execution systems: An interdisciplinary challenge. In Proceedings of the 2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010), Bilbao, Spain, 13–16 September 2010.
55. Witsch, M.; Vogel-Heuser, B. Formal MES Modeling Framework—Integration of Different Views. *IFAC Proc. Vol.* **2011**, *44*, 14109–14114. [[CrossRef](#)]
56. Witsch, M.; Vogel-Heuser, B. Towards a formal specification framework for manufacturing execution systems. *IEEE Trans. Ind. Inform.* **2012**, *8*, 311–320. [[CrossRef](#)]
57. Weißenberger, B.; Flad, S.; Chen, X.; Rösch, S.; Voigt, T.; Vogel-Heuser, B. Model driven engineering of manufacturing execution systems using a formal specification. In Proceedings of the 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), Luxembourg, 8–11 September 2015.
58. Flad, S.; Weißenberger, B.; Chen, X.; Rösch, S.; Voigt, T. Automatische Generierung von Fertigungs-Managementsystemen. In *Handbuch Industrie 4.0 Bd. 2*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 349–368.
59. Chen, X.; Gemein, F.; Flad, S.; Voigt, T. Basis for the model-driven engineering of manufacturing execution systems: Modeling elements in the domain of beer brewing. *Comput. Ind.* **2018**, *101*, 127–137. [[CrossRef](#)]
60. NAMUR. *NA 110—Benefits, Design and Application of MES*; NAMUR: Leverkusen, Germany, 2006.
61. Drath, R. Die Zukunft des Engineering: Herausforderungen an das Engineering von fertigungs- und verfahrenstechnischen Anlagen. In Proceedings of the Tagungsband Karlsruher Leittechnisches Kolloquium, Karlsruhe, Germany, 28–29 May 2008.
62. Naedele, M.; Chen, H.M.; Kazman, R.; Cai, Y.; Xiao, L.; Silva, C.V.A. Manufacturing execution systems: A vision for managing software development. *J. Syst. Softw.* **2015**, *101*, 59–68. [[CrossRef](#)]
63. FoodDrinkEurope. Data & trends of the European food and drink industry. In Proceedings of the Confederation of the Food and Drink Industries of the EU, Brussels, Belgium, 11 October 2018.
64. Chen, X.; Voigt, T. Implementation of the Manufacturing Execution System in the Food and Beverage Industry. *J. Food Eng.* **2020**, *278*, 109932. [[CrossRef](#)]
65. Kather, A.; Voigt, T. *Weihenstephan Standards for the Production Data Acquisition in Bottling Plants: Part 1: Physical Interface Specification Part 2: Content Specification of the Interface Part 3: Data Evaluation and Reporting Part 4: Inspection and Safe Operation*; TUM, Lehrstuhl für Lebensmittelverpackungstechnik: Freising, Germany, 2010.

66. Bär, R. Weihenstephaner Standards for Brewing Processes. In Proceedings of the Kick-Off-Meeting, Weihenstephan, Germany, 14 February 2017.
67. Kunze, W. *Technologie Brauer und Mälzer*, 11th ed.; Überarbeitete Auflage; VLB: Berlin, Germany, 2016.
68. Willaert, R.G.; Baron, G.V. Applying sustainable technology for saving primary energy in the brewhouse during beer brewing. *Clean Technol. Environ. Policy* **2004**, *7*, 15–32. [[CrossRef](#)]
69. Muster-Slawitsch, B.; Weiss, W.; Schnitzer, H.; Brunner, C. The green brewery concept—energy efficiency and the use of renewable energy sources in breweries. *Appl. Therm. Eng.* **2011**, *31*, 2123–2134. [[CrossRef](#)]
70. Bär, R.M.; Voigt, T. Analysis and Prediction Methods for Energy Efficiency and Media Demand in the Beverage Industry. *Food Eng. Rev.* **2019**, *11*, 200–217. [[CrossRef](#)]
71. Scheller, L.; Michel, R.; Funk, U. Efficient use of energy in the brewhouse. *MBAA TQ* **2008**, *45*, 263–267.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).