9th International Conference on Mass Customization and Personalization – Community of Europe (MCP - CE 2020)
Re-innovating Business in the Digital Era
September 23-25, 2020, Novi Sad, Serbia

INTEGRATING ARCHITECTURAL DESIGN CHANGES IN COMPUTER-AIDED DESIGN OPTIMIZATION

Philipp Wolniak, Daniel Kloock-Schreiber, Bastian Sauthoff, Roland Lachmayer
Leibniz University Hannover, Institute of Product Development, Hannover, Germany

Abstract: *One opportunity in product development is to customize the product depending on the customer use-case scenarios rather than using static product requirements. This is facilitated by the increasing capabilities of computer-aided design. However, the necessary formalization of the early design stage analysis-synthesis routines for an automated optimization to the use-case emerges as a challenging problem. To address this problem, an approach is presented which combines an optimization framework with a functional simulation model and a generative design approach (GDA) model. The GDA is based on the combination of a parametric geometry model and the generative exchange of parts of the model. While the functional model evaluates the objective function at hand, the GDA model ensures the physical feasibility.*

Key Words: design automation, optimization, generative design approach

1. INTRODUCTION

Product developments are driven by the fulfillment of a set of requirements, either strictly specified by a customer or through market analysis and the anticipation of the customer's needs. Specified requirements often restrict the development and do not represent the optimal solution. A customization of a product to the customer's needs upon a user-scenario analysis gives the developer a higher degree of freedom in the realization of a scenario-specific solution. However, it also imposes a more dynamical and flexible necessary reaction to e.g. market situation changes or new customer needs [1, 2]. This leads to a high spectrum of possible solutions to user-scenarios tailored to the customer. Finding a solution to this kind of problem comes down to the balancing of what kind of respective solution meets the specific needs in which defined way [3].

Typically, in product development, this is performed in an iterative process, where a set of characteristics of the potential solution is analyzed and the properties are compared to the requirements. This process is repeated until a suitable match for the requirements is found [4]. As even small geometric or functional changes can have a strong influence on these properties this design task can lead to a high number of necessary model changes and analysis routines. The actual transition of changes and model information into simulation tools for system design currently still relies on laborious manual transfer. The user himself has to perform model changes or analytical calculations, pass the models to Computer-Aided Engineering (CAE) tools like 3D modeling environments (e.g. CAD), Finite Element Analysis (FEA), or functional performance analysis. Therefore, the need for a new variant or changes to a product also leads to a high variety of these CAD and analysis models. A possibility to overcome this is the use of design automation, e.g. in the form of parameterized models. The reason why these tasks are still typically performed manually is the effort required to plan the parameters, their dependencies and the corresponding model structure in a parameterized model. Especially the connection of functional properties to geometric characteristics and the imposed restrictions results in a high managing effort without proper planning. Therefore, the more complex the geometry and the larger the assemblies, the more important it is to constrain model parameters and reference individual features to build robust CAD and analysis models [5–7].

Modeling languages like UML or SysML aim to give the user the possibility to model functions and their solutions [8]. Approaches like graph-based languages [9] even go as far as to implement geometric properties into the formalized modeling language and build-up a complex network of dependencies. However, these modeling languages either mainly focus on the product structure and only abstractly link sub-assemblies or parts with the actual functional outcome, or they are based on a discrete network which is rather inflexible upon changes. Thus, these modeling languages lack a detailed description of how the geometric model and the associated dependencies have to be built and connected to support an increasing design automation.

The aim of the research in this paper is to present a method for the build-up of a robust geometry model, capable of adapting to necessary product variants in the iterative development cycles. Furthermore, the transfer of model parameters to a functional simulation environment inside an automatic product optimization is given. The proposed method is based on the generative design approach (GDA) as a means of a model-based integration of constraints and dependencies. This approach uses a parametric geometry model as a template and gives the possibility to insert specific elements, adapting robustly to the template characteristics. The method is shown using an example of a business-to-business development of a coffee machine for high throughput. This coffee machine is specifically designed for the user-scenario of a hotel.

2. THEORETICAL BACKGROUND

In the product development the search for a solution, matching the requirements is a search in a distinct solution space. Whereas, a solution space describes all theoretically possible solutions for a task or set of requirements [10]. The beforehand described iterative process of analyzing characteristics and comparing the properties to the requirements is an exploration of this solution space. Several established approaches like the development process of the German VD2221 [11] use this idea throughout the stages of the development. Using a functional structure, the overall problem is sub-divided into smaller parts and related to sub-functions. The solution finding consists of defining single artifacts fulfilling the sub-functions and combining them.

Those typical approaches can be described as process-oriented, giving a processual way of the development, without a detailed description of the actual modeling work [12]. Another approach, which reverts this understanding is presented by Weber [5] consisting of the Characteristics-Property Modeling, which describes the product modeling part and a separate process model, the Property-Driven Development. This distinction is used in the analysis-synthesis routines used to update both models to the given design problem. Gradually, throughout these routines properties of a certain set of characteristics are adapted. By explicitly addressing the use of computer-aided engineering tools and by defining the product development as a mathematical optimization problem, a certain stage of automation is supported [12, 13].

A more functionally centered approach is given by Gero [14] with the Function-Behavior-Structure (FSB) modeling. This starts with the requirements (R), which are transformed in a first process into a function state space (F) and further into a behaviour state space (Be). Based on this, a structure (S) is synthesized, which is analyzed and a behaviour (Bs) is derived. In an evaluation (Bs) and (Be) are compared and the structure is documented (D), if a reformulation is necessary, the structure, the behaviour, or the functional state space is modified [14].

The Axiomatic Design method is based on a domain concept in which a distinction is made between the customer domain (attributes the customer is looking for), functional domain (functional requirements and constraints), physical domain (design parameters representing a design solution) and process domain (parameters and tolerances of the manufacturing process). The domains structure the development process and lead to iterative jumps between adjacent domains until a solution can be assigned exactly to a requirement [15].

In contrast to solution spaces, Gero [16] describes variation spaces in which known constructions can be adapted to new or changed requirements. Routine Design is used to describe activities in which parameters are adapted. Variation and adaptation constructions are described with Innovative (retaining material, function and essential form) and Creative Design (retaining the solution principle) [16]. Thus, Gero already postulated the use of parametrics and templates in the early 1990s [7].

System Dynamics (SD) is a tool to capture feedback processes, inventories and flows, time delays and other sources of dynamic complexity in systems. It supports the design and evaluation of new system structures and their consequences [17]. For this purpose, an SD model consists of in- and outflows that are controlled by valves and connect stocks. Stocks outside the system boundaries are integrated as sources or sinks. Further essential and linking elements of the model are control loops and parameters [18].

In the approach of Kloock-Schreiber et al. [19], for the modeling of PSS solution spaces with SD, the main function of the PSS with the corresponding flows and control loops is first described. This is followed by the detailing, for the entire system, or in sub-models limited to individual system areas.

While these approaches describe mainly a processual way of the exploration of the solution space, other approaches give a clearer description of the actual modeling work. A basic way for the automatic creation of variants is the use of the parametric design, where the ability of a CAD program to define parameters and combine them via constraints is used [20]. A parametric CAD model can represent different variants and is, therefore, able to span a solution space [21]. To build a solution space the knowledge must be explicitly implemented in digital prototypes. Besides the parameters of the models, mathematical and logical boundary conditions and constraints can be defined between them. Thus, a solution space is described by the designer by defining not only the product shape but also the variant design and the associated control and configuration concept for the components [20, 21]

The defined parametric model requires the logical understanding of the geometry or assembly at hand and the transfer of this understanding into the parameters and constraints. While this does relieve the designer of repetitive tasks, it also means a high amount of model preparation and adjustment to the own problem [22].

This is taken from the designer in the knowledge-based approaches, where the product knowledge is stored beforehand and then generically used afterward [23–25]. A common way of automating the CAD model build-up and change is the definition of subparts of a geometry. La Rocca and van Tooren [26] propose the use of high-level primitives inside a knowledge-based system. These primitives are highly adaptable to the adjacent geometry of other primitives when completing the model. The proposed framework is based on an object-oriented approach with a code-based integration of these primitives.

Amadori et al. [27] use a similar approach, called "High-Level CAD templates" with the distinction of the creation and utilization inside a CAD environment. The method is shown in different use cases, ranging from a robot model to airplane design. Another approach is given by Schmidt and Rudolph [9] where a graph-based design language is used to automatically initiate models, which are usable in a high number of analysis programs. The idea is the abstraction of model parts and dependencies to a generic level, using e.g. UML as a modeling language. Via a compiler, these graphs are then translated back, with the possibility of generating a high number of variants in different respective model environments.

Another approach based on the subdivision of the geometric entity is the generative design approach (GDA) [28] which uses high fidelity model parts to combine these in an assembly model. These parts are adaptable to the surrounding via the information and dependencies stored internally in the parts and in interfaces connecting these parts. The advantage is the separation of local dependencies into the respective submodels and the limitation of global dependencies.

What all of these approaches have in common is that the geometry model and its underlying properties are built to fulfill an objective function, mainly dependent on the geometry like the weight, mechanical stresses or a flow optimization. Although functional characteristics also may be an objective, this is only expressed through the geometric continuity of the model. The advantage of this continuity is the possibility of clearly defining the interfaces between the subparts and therefore coherently transferring information upon model build-up. There is nevertheless the lack of a clear description of how to model a functional assembly, where the geometric continuity between subparts of the model is not given. Besides, the use of a defined constraint network or rules as a basis for the implementation of constraints affects the model preparation, as every possible combination of parameters and its outcome has to be planned.

Therefore, in this paper the GDA is used as a basis and adapted to the modeling requirements of a functional assembly. The model-based approach gives the advantage, of implementing every constraint in one subpart. The overall constraint network is built and checked automatically upon the merger in the product assembly. Besides, the parametric basis of this approach is suitable for use in a numerical optimization.

3. METHOD

This chapter introduces the method for the build-up of a robust geometric model for the use in the design automation and especially in a numerical optimization routine. Typically, this poses a major challenge in parametric CAD modeling when used throughout parameter variations with a high range of possible combinations. If a certain parameter combination results in a crashed CAD-model, no property values of the model can be used for the objective function of the optimization. The whole process is stopped. However, this robustness is not only of importance regarding an error-free build-up but also for the logical connection of subparts and the resolution of dependencies between these components.

For a better understanding of the proposed method, the example of a coffee machine for high throughput is given throughout this chapter. The internal components and their functional behavior are modeled and later on optimized for a use-case of a customer. The basic structure of the machine consists of a brewing unit from which the coffee is pumped to a buffer tank. This minimizes waiting time, as the disposal of coffee and the brewing of new coffee can take place at the same time. A further schematic description of the components of the machine is given in Fig.2.

3.1 Framework for the automated Optimization

In this paper the underlying optimization process as presented in Fig. 1 is used.

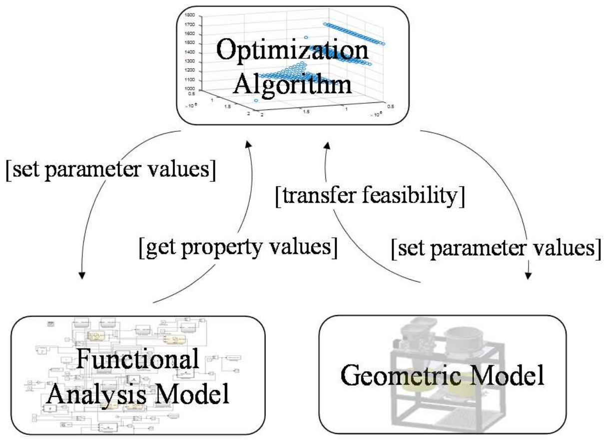

Fig. 1: *Framework for the optimization*

For a comprehensive concept evaluation, the change in dimensional sizes of the product and its components has to be linked with the resulting functional outcome.

Therefore, a functional model is used, which simulates the time-dependent behavior of the product. The geometric changes are formalized in a geometric CAD model. To interlink these models and ensure a correct parameter transfer an optimization algorithm is used as the leading instance. This algorithm changes the parameters inside the restricted design space. The sequence of parameter changes, as well as the model update, is determined in the optimization algorithm. Every further synthesizing and simulation step is within the functional analysis model and the geometric model and thus completely separated from the optimization process. The advantage is a transparent modeling process and a comprehensible solution-finding.

3.2 Method for the build-up of a robust geometric assembly model

One approach to establish a robust parametric CAD-model is the clear definition of constraints within the model. Especially by using geometric constraints complex constraint equations can be built inside the model. If the constraint solver is not able to solve the equations for a defined set of parameter values or if there are several solutions, where some of them result in a non-logic geometric output, the model can't be regenerated.

For the coffee machine and the framework presented in this paper, the clear definition of constraints is established using a GDA model. This is based on the definition of so-called design sections as a logical skeleton for the parametric model. These design sections are derived from the functional structure of the analysis model and describe the related geometric entities independent of the product components. Instead of defining a section for every component or coherent assembly itself, the definition of a design section may also virtually cut through a component. This approach is visualized in Fig. 2, where the design sections for the coffee machine are shown. An example of this virtual cut is the water container section, which consists mainly of the water container and the interlinking pipes from the pumps. Therefore, the overall height of this design section is dependent on the height of the water container and the space necessary to store the pipes segment.

The positioning of these design sections and the logical structure of the skeleton is used to represent different product structures. However, it has to be noted, that the change of a product structure inside one single optimization routine raises the model complexity, due to a higher number of dependencies. Additionally, the change of the product structure in the form of a skeleton modification imposes a set of further optimization parameters and therefore also increases the complexity of the solution space. Thus, the approach in this paper is to perform several parallel optimization routines with varying structures. This brings the advantage of a plainer solution space and the absolute comparison of the respective best solution value.

The consistent merging of the design sections in the skeleton is mainly driven by the interface definition. The idea of this interface definition can be seen in Fig. 2 with the description of the functional interface openings, represented by the light squares or circles and the black arrows, schematically describing the functional flow through the assembly.

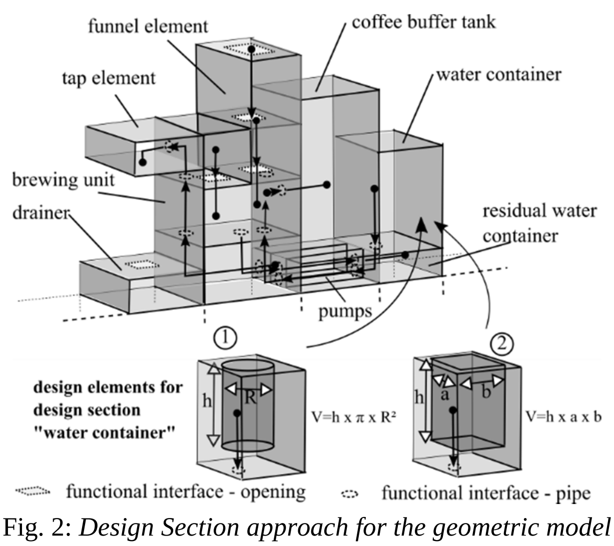

Fig. 2: *Design Section approach for the geometric model*

This flow again is derived from the energy and material flow through the product. In this exemplary case, this is represented by the water flow, from the container to the brewing unit and the tap element, as wells as the coffee flow from the dosing unit to the brewing unit. Therefore an interface is given, where on a logical basis flow from one design section to another is provided.

This approach allows defining in which way the underlying components should interact with each other without adding constraints directly to every entire component. Also for the definition of parameter value limits, this approach provides a framework: There are some absolute lead parameters for the global parameters of the model like height, length or width, for which an absolute limit can be specified. All other parameters for the size of the inner design sections are defined as relative parameters (value range 0…1) in reference to the lead parameters. Every design section as placeholder has a representing geometric entity. These geometric entities are so-called design elements, which are parametric CAD-parts. As described before, these CAD-parts are not bound to represent a complete product component, but may also include virtually cut parts of other components. The parameters and constraints from the design section definition are used in these design elements as a template for parameter referencing. The design element inserted into the design section, therefore, inherits the parameter definition of the design section. All parameters of a design element are relative parameters that are (directly or indirectly) linked to the parameters of a design section. By this, a defined constraint definition is also guaranteed for the underlying geometric entities.

Furthermore, there can be provided more than one design element for every design section as shown in Fig. 2 with the two different design elements for the water container. These design elements have to be designed beforehand and stored in a design library. This library can later be used to insert a variety of different design elements and therefore alter the actual conceptual design. Hereby, conceptual changes of the model are supported, too.

In a first step, this model can be used as a means of a preliminary design, where the design elements in the design sections are represented as simple basic geometries, like cuboids or cylinders, constituting the necessary building room. In the further development process, when constantly adding design elements with the actual design, a more detailed model of the product can be built automatically from the design section templates.

3.3 Design section and interface definition

The following section gives an overview of the methodical derivation of the parameters and the design sections leading to the assembly model. Fig. 3 shows the basic methodical process. The first step is to define the functional structure of the product at hand. Using this functional structure as well as the energy and material flow a defined dependency structure of all components of the assembly is given. In addition, the functional structure gives a formalized and comprehensible way of defining those components, which have properties that affect the objective function (step two). These components and their parameters are later on used as the main drivers of the optimization. As the objective function and its outcome for a specified parameter variation is computed by the functional analysis model, the third step is to create this model. Depending on the objectives of the development and the product at hand, this model must be able to reflect these objectives.

In this exemplary case of the coffee machine, three main objectives that define the development are chosen. The first objective is the working time efficiency of the machine, regarding the serving of cups of coffee. The second objective is water consumption and therefore the efficiency of the brewing and buffering process. The third objective is the costs of the overall coffee machine, as this is a key measure for the profit of the producing company. Therefore, a time-dependent discrete event model in Matlab/Simulink is used to analyze the complete brewing, pumping, buffering and serving process [29]. For a parameter variation the main parameters, like the diameter of the brewing unit and the buffer tank, the pump sizes or the pipe sizes are chosen. As these parameters have a direct impact on the volumetric flow, they affect the objectives one and two, and because of the component changes also objective three.

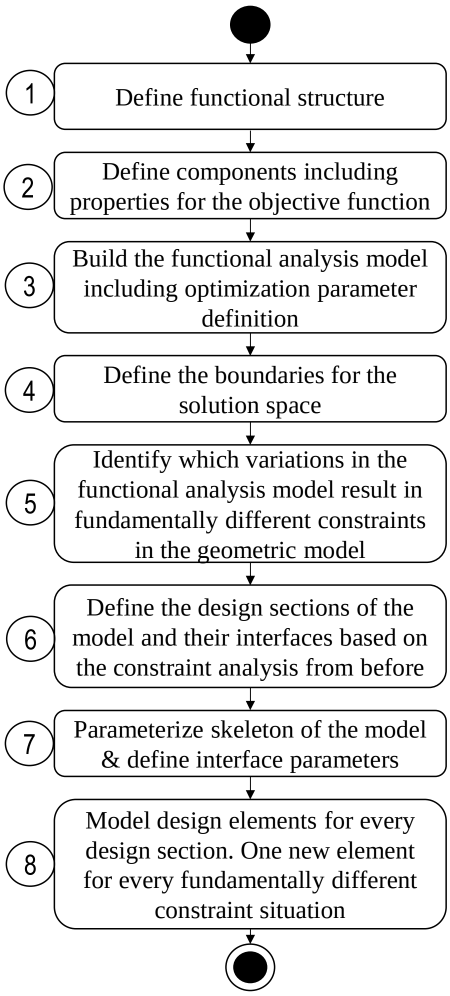

Fig. 3: *Method for the geometric design section model*

In the following step, the boundaries for the solution space of the model are defined. This is mainly done by giving an upper and lower boundary for the parameters from step three. In this example, the boundaries are defined by the manufacturing restrictions. A distinction between components which are continuously changeable (e.g. the brewing unit which is manufactured in an in-house sheet-metal deep-drawing process) and discretely changeable components (mainly commercial off-the-shelf parts, like pumps or motors) is made. Thus, the boundaries can either be chosen according to the manufacturing possibilities or via assigning a value range and therefore a discrete variation of a specific component.

Regarding the geometric model, the main activity is to define how many design sections are necessary, where these design sections are placed and which parts of the actual geometry are inside every section. A preparation for this is the fifth step, where an analysis is done for which variation in the functional analysis model results in a fundamentally different constraint in the geometric model. This analysis is necessary to lower the necessity of implementing value-dependent constraints, which later on have to be managed externally. After this, in step six, the design sections, the components inside these sections and the interfaces between the sections are defined. The outcome for the exemplary case can be seen in Fig. 2 and is derived as already described in 3.2.

After defining the design sections and the optimization parameters the basic structure of the model is given. The following step is the integration of the design sections in a global model which is defined by the design skeleton. For this integration the definition of the interfaces becomes crucial. As described in 3.2, the functional structure is used to determine the flow through the design sections. When combining the sections, the question arises which design sections have to be combined statically and which design sections may change independently.

In a static combination, the geometric change of a design section is transferred directly to another design section by adjusting its position in the assembly. This is shown schematically in Fig. 4, where mainly the design sections of the brewing unit, the motor for the brewing unit and the coffee disposing unit are in the focus.

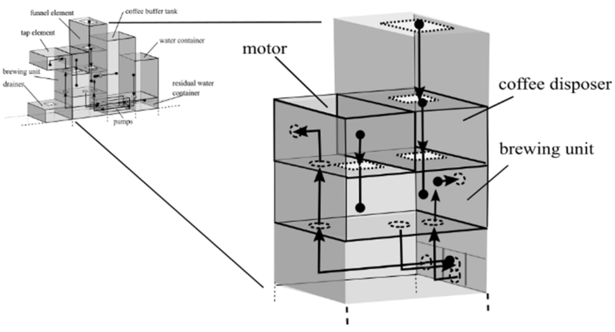

Fig. 4: *Overview of design sections dependencies*

As indicated by the black arrows and the functional interface openings between the design sections, a functional connection is given between the design section of the brewing unit and the motor design section, as well as, between the design section of the brewing unit and the disposing design section. The connection between the motor and the brewing unit is given because of the energy flow of the motor torque to the brewing screen via a toothed rack. The connection between the coffee disposer and the brewing unit is given by a material flow of the coffee from the disposer to the brewing screen. Now, a parameter change, like the height of the brewing unit requires a static connection to the other two design sections. This preserves a coherent global model.

Therefore, a change in height of the brewing unit requires a change in the vertical position of the other design sections. On the other hand, a change in the brewing unit may functionally also lead to the necessity of a more or less powerful and therefore larger or smaller motor. This functional change may also imply changes in the size of the motor design section, due to the exchange to a new motor version. In this scenario, while the two design sections of the motor and the brewing unit adjust their position interdependently, the design section of the coffee disposer only adjusts its position according to the interface with the brewing unit. As there is no flow between the motor and the coffee disposer, these design sections do not have to adjust accordingly.

Changing the size of the motor design section may, therefore, lead to a gap between the design section of the motor and the coffee disposer. This still defines a feasible solution, because no functional or dependent intersection is given. The contrary case of a greater design section of the motor unit may lead to an intersection. Since this implies that there is not enough space for the components in the selected combination, this solution is considered infeasible and is returned as such to the optimizer.

After defining the design sections and their behavior in the global model, the seventh step is to parameterize the skeleton and define the interface parameters. As shown in Fig. 4, the transparent passages and their position on the interfaces of the design sections play a vital role. Therefore the interface between two or more design sections is parameterized. Fig. 5 gives an exemplary overview of the interface parameterization between the design sections of the motor, the coffee disposer the and brewing unit.

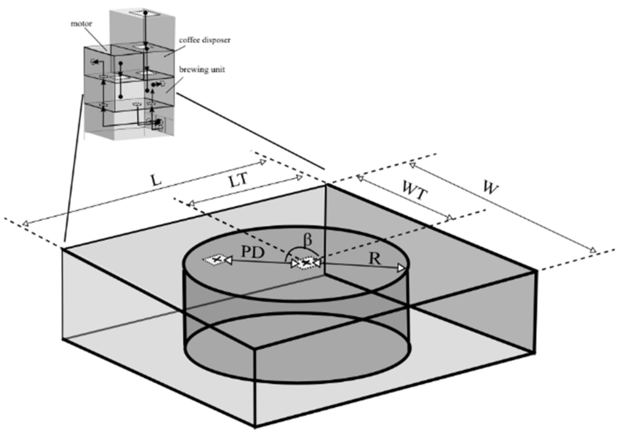

Fig. 5: *Design section interface parameterization*

The parameter of the tank radius in the brewing unit is used as an optimization parameter, as it mainly affects all three objective functions. Therefore this parameter is integrated as a leading parameter into the interface parameterization. The functional interface openings are used for the positioning of the adjacent components in the other design sections. To prevent a collapsing of the design section because of an intersection of the component and the design section the following equations are used:

$$LT = (p_1 * (L - 2 * R)) + R \tag{1}$$

$$WT = (p_2(W - 2 * R)) + R \tag{2}$$

$$PD = p_3 * (R - 2 * l_{min}) + l_{min} \tag{3}$$

The position of the center of the brewing unit in the design section is controlled by the relative parameter p_1 and p_2 in the equations (1) and (2). Due to the relative representation regarding the parameters L, representing the length of the design section, and W representing the width of the design section, an overlap is avoided. Equation (3) controls the position of the functional interface opening and therefore the position of the disposing component in the disposing design section. By adding the constant l_{min} a sufficient clearance between the center point of the disposing component and the outer and inner barrier of the brewing unit is given. An extreme value analysis shows, that no inference of the disposing unit opening and the tank radius is possible:

$$PD(p_3 = 0) = l_{min} \tag{4}$$

$$PD(p_3 = 1) = R - l_{min} \tag{5}$$

The same applies to the positioning parameters of the center point of the tank. Additionally, the parameter of the angle β is used to vary the position of the center point of the disposer component. The last step is to define the design elements that are inside the design sections. Every design element interprets the interface parameters differently. The interface parameterization can be seen as a template that the design elements adjust to. Therefore in this step a design element library is created, where every new instance of the design element can be stored and later on reused. A new instance of the design element may be e.g. a component manufactured using different manufacturing processes. In this example, the interface parameter R from Fig. 5 defines the Radius of the brewing chamber in the brewing unit. The actual form of the chamber is defined in the design element itself.

4. CASE STUDY

The case study is referring to the example of the coffee machine concept, as shown schematically in Fig. 2. A functional model describing the brewing process and the transfer of the coffee from a brewing unit to a buffer tank and finally to a tap element is used for the functional analysis. Inside this functional model, every component and its parameters (functional and geometric) are implemented and therefore define the outcome. These parameters range from the geometric sizes of the brewing unit and the buffer tank to functional parameters like the volumetric flow of the pumps or the power output of the flow heater. As the aim is to show the impact of the presented method on the geometric model and its robustness throughout the optimization, this case study is performed using only one skeleton and therefore one product structure. Additional product structures are modeled later and can then be used to compare respective optima.

The metrics to quantify the coffee machine in this case study are the objective functions of the working hours and water consumption. Two cases are displayed: Firstly, a single objective optimization of the performance of the machine, measured with the working hours. This optimization gives a good representation of the outcome of changes in the objective function and the convergence of the design parameters. Secondly, a multi-objective optimization with the performance of the machine and water consumption to display a realistic consideration of the functionalities and properties in the product development.

4.1 Single-objective optimization

The single objective optimization is performed using the Particle Swarm optimization algorithm [30]. Fig. 6 gives an overview of the user-scenario used to perform the optimization.

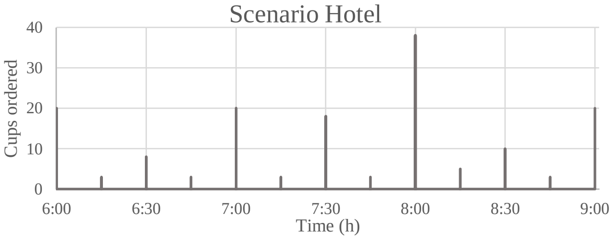

Fig. 6: *User-scenario of a hotel in the period of three hours*

The chosen scenario is a period of three hours in the morning in a hotel, where at several peak times guests have to be served. The metric of the optimization in this example is the working hours of the machine until every customer is served. Fig. 7 depicts the optimization process.

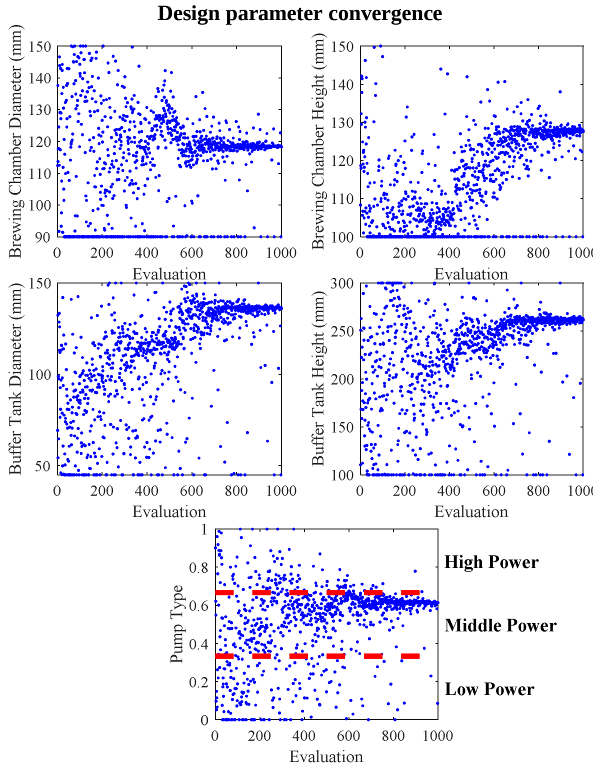

Fig. 7: *Optimization parameters convergence*

The five graphs show the values of the parameters of the brewing chamber diameter and height and the buffer tank diameter and height according to the number of evaluations, as well as three discrete pump types with different power outputs. For the discrete power pumps a value range is chosen. The boundaries are defined from 0 to 1. If the optimizer chooses a value between 0 and 0.33 the low power pump is chosen and so on. As can be seen, the optimization algorithm converges with defined values regarding the parameters after about 1000 evaluations.

The upper graph in Fig. 8 shows the value of the objective function of the working hours over the number of evaluations. Additionally, every unfeasible parameter combination, due to intersecting design sections, is marked with a cross at the top in this graph. These intersecting combinations are flagged by the geometric model and passed to the optimization algorithm.

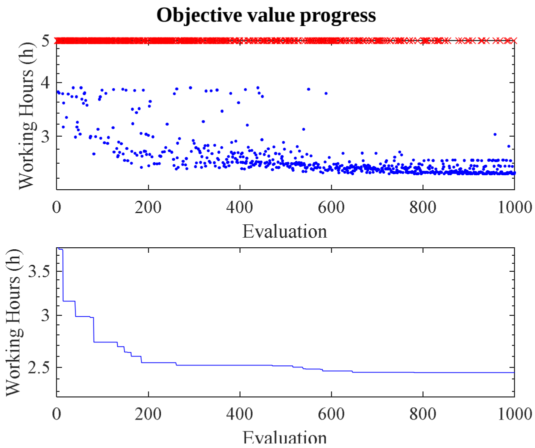

Fig. 8: *Change of the Objective Function over the evaluations*

Due to the violation of the restriction as a part of a death penalty, these solutions are not considered in the solution-finding. As can be seen, the number of crosses is declining towards the end of the optimization. The lower graph in Fig. 8: *Change of the Objective Function over the evaluations*Fig. 8 depicts the change in the best objective function value over the number of iterations. After about 200 evaluations the best objective function value declines rapidly. In the following 300 evaluations, this value stays steady, followed by several minor changes at about 500 evaluations.

This second change correlates with parameter changes of the brewing chamber in the brewing unit in Fig. 7. Accordingly, at about 500 evaluations the diameter and height of the brewing chamber are significantly increased, leading to the improvement in the best objective function value. Another point to be marked in this evaluation is the convergence of the pump type to the middle power pump. As this is a single objective optimization, there is no conflicting objective function, as e.g. the cost, where a higher power pump alters the outcome with a higher price. Therefore a higher power pump leads to a lower working hour and a better objective function. The fact that the optimization algorithm converged to the middle power pump comes from the geometric restriction of the overall height of the machine. Fig. 9 gives an overview of the geometric model used for the optimization, displaying only the design sections without the design elements inside.

A transparent enclosure marks the overall machine geometry restrictions for the height, width and depth. The figure depicts the status of a geometric violation of the restrictions. The brewing unit, buffer tank and adapted to this the water container, are sized according to the converged parameter values from Fig. 7. The dotted line in the lower-left corner marks the difference in the geometric size of the middle and high power pump design section. The increase in volumetric flow regarding the high power pump does not compensate for the lower necessary tank volumes to fit the high power pump in the overall concept.

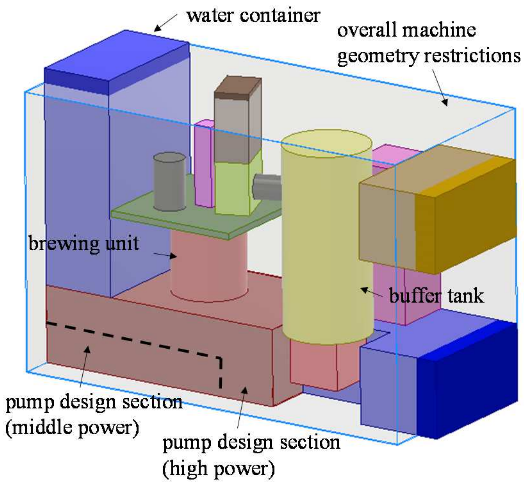

Fig. 9: *Restrictions in the geometric model*

This can also be visualized in the objective function values. A three-dimensional view of the parameters of the buffer tank and the objective function value (Fig. 10) shows that there is a hard restriction in the solution space, marked with a vertical dotted line. For a better understanding, the objective function values which were determined geometrically unfeasible have been put to the top of the diagram. An accumulation of solution points around the objective function value of 2.5 working hours and the design parameter values of 140 mm in diameter and 250 mm in height can be seen around the found optimum. A further increase in the height or the diameter of the buffer tank, although leading to a better objective function value, is discarded. The geometric model, therefore, prevents the optimization algorithm from stepping out of the restricted boundaries.

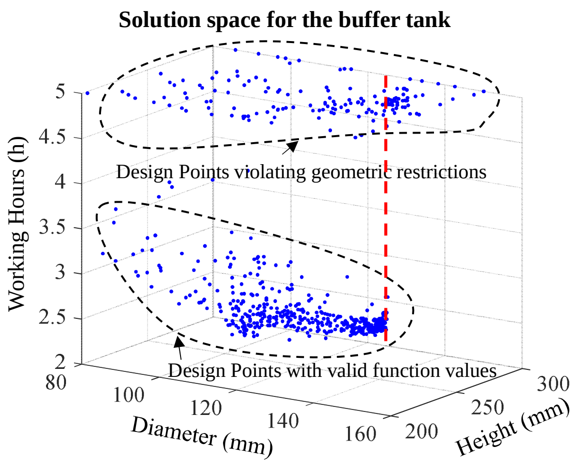

Fig. 10: *Influence of restrictions from the geometric model on the solution space*

4.2 Multi-objective optimization

The multi-objective optimization is performed using the Non-dominated Sorting Genetic Algorithm II [31]. Again the scenario as presented in Fig. 6 is used for the optimization. Fig. 11 depicts the objective value space after 2500 iterations. The water consumption of the machine throughout the iterations is marked on the y-axis, while the performance quantified by the working hours is marked on the x-axis.

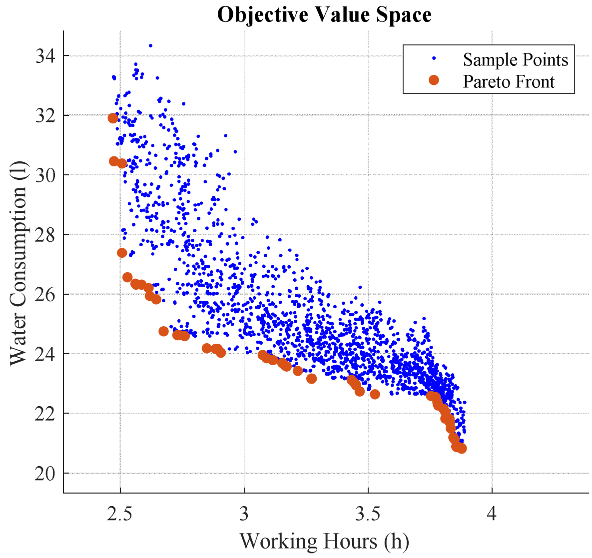

Fig. 11: *Objective value space in the multi-objective optimization*

The bigger dots mark the Pareto-front indicating the parameter combinations where any further change automatically leads to a higher value of one of the respective objective functions. Although the front is not fully formed, the result still is a coherent representation. Comparing this to the result from the single-objective optimization, the optimum from the first optimization can be found on the far left side of Fig. 11. As can be seen, starting from the value of 2.5 working hours to the left there is a steep increase of the second objective function. A slight improvement of the performance comes at the cost of exponentially higher water consumption. In product development, this outcome can be used for decision making by weighting the objective functions. This can be done according to the customer and their preferences regarding performance or efficiency.

5. CONCLUSION AND FUTURE WORK

This paper proposes a method based on the generative design approach (GDA) for the build-up of a geometric CAD model, capable of robustly representing the geometric solution space inside a numerical optimization routine. By using design sections and their main defining parameters instead of the fully detailed component models the modeling effort is reduced, while maintaining the necessary extent to give a precise outline of the geometric feasibility. The advantage of the proposed method is a structured procedure for creating a robust geometric model. By using the functional structure as a starting point the geometric changes are directly linked to the implied functional changes, making it directly usable in a functional analysis as described in the presented case study. Additionally, by implementing all geometrical constraints and restrictions into the geometrical model, and furthermore into every single design section, the complexity of the constraint handling is reduced.

In the presented case study of the optimization of a coffee machine the use of the geometric feasibility is shown, by effectively restricting the solution space. While a single-objective optimization of the performance of the coffee machine shows the properties and the outcome on the objective function, a multi-objective optimization additionally gives insight into a realistic transfer of the approach into the product development. The found optimal solutions are used as a starting point for decision making. The geometric model with the respective design sections gives a template for the more detailed models of the internal product components. Using a design element library the time for modeling a detailed version of the coffee machine is reduced.

It is however to be noted, that the outcoming geometric model of the found optimum still has to be post-processed, according to the grade of detail chosen in the beginning to match the actual components to the found main parameters. Also, while most of the necessary information is given through the initial development in the conceptual phase, the derivation of the design sections, the interface definition and the parameterization needs preparation and adaptation to the problem at hand. Furthermore, the initiation of the necessary design elements and the consistency of the design element library are aspects, which need preparation and work throughout the development. Future work lays in the detailing of the design element library and the integration of detailed models into the optimization routine to lower the post-processing time.

6. ACKNOWLEDGMENT

This research was conducted in the scope of the research project SmartHybrid – Product Engineering (ID: 85003608) which is partly funded by the European Regional Development Fund (ERDF) and the State of Lower Saxony (Investitions- und Förderbank Niedersachsen NBank). We also like to thank the Institute of Structural Analysis from the Leibniz University Hannover for the provision of the optimization framework.

7. REFERENCES

[1] H. Meier, R. Roy and G. Seliger, *Industrial Product-Service Systems—IPS 2*. *CIRP Annals*, vol. 59, no. 2, pp. 607–627, 2010

[2] D. Schreiber, P. Gembarski, R. Lachmayer, *Developing a Constraint-Based Solution Space for Product Service Systems*, 8th International Conference on Mass Customization and Personalization - Community of Europe (MCP-CE 2018), 2018

[3] A. Chakrabarti and T. P. Bligh, *A scheme for functional reasoning in conceptual design*. *Design Studies*, vol. 22, no. 6, pp. 493–517, 2001

[4] D. C. Wynn and C. M. Eckert, *Perspectives on iteration in design and development*. *Research in Engineering Design*, vol. 28, no. 2, pp. 153–184, 2017

[5] C. Weber, Design Theory and Methodology - Contributions to the Computer Support of Product Development/Design Processes," In: BIRKHOFER, Herbert (Hrsg.). *The Future of Design Methodology*. London : Springer London, 2011, pp. 91–104

[6] H. Li, P. C. Gembarski and R. Lachmayer, Template-Based Design for Design Co-Creation," In: *Proceedings of the 5th International Conference on Design Creativity (ICDC2018)*, 2018

[7] P. C. Gembarski and R. Lachmayer, *Solution Space Development: Conceptual Reflections and Development of the Parameter Space Matrix as Planning Tool for Geometry-based Solution Spaces*. *International Journal of Industrial Engineering and Management*, vol. 9, no. 4, pp. 177–186, 2018

[8] G. Booch, *Object-oriented analysis and design with applications*. Upper Saddle River, NJ, Harlow, Addison-Wesley, 2007

[9] J. Schmidt and S. Rudolph, *Graph-Based Design Languages: A Lingua Franca for Product Design Including Abstract Geometry*. *IEEE computer graphics and applications*, vol. 36, no. 5, pp. 88–93, 2016

[10] J. Ponn, Systematisierung des Lösungsraums," In: LINDEMANN, Udo (Hrsg.). *Handbuch Produktentwicklung*. München : Hanser, 2016, pp. 715–742

[11] VDI, *VDI2221 - Design of technical products and systems* : *Model of product design*. 2019

[12] B. Sauthoff: *Generative parametrische Modellierung von Strukturkomponenten für die technische Vererbung*. Leibniz Universität Hannover; TEWISS - Technik und Wissen GmbH. Dissertation. 2017

[13] C. Weber and H. Werner; Universität Des Saarlandes (Mitarb.)*Klassifizierung von CAx-Werkzeugen für die Produktentwicklung auf der Basis eines neuartigen Produkt- und Prozessmodells*, Design for X: Beiträge zum 11. Symposium, 2000

[14] J. S. Gero and U. Kannengiesser, The Function-Behaviour-Structure Ontology of Design," In: CHAKRABARTI, Amaresh; BLESSING, Lucienne T. M. (Hrsg.). *An Anthology of Theories and Models of Design*. London : Springer London, 2014, pp. 263–283

[15] N. P. Suh, *Complexity in Engineering*. *CIRP Annals*, vol. 54, no. 2, pp. 46–63, 2005

[16] J. S. Gero, *Design Prototypes: A Knowledge Representation Schema for Design*. *AI Magazine*, vol. 11, no. 4, 1990

[17] J. D. Sterman, *System Dynamics Modeling: Tools for Learning in a Complex World*. *California Management Review*, vol. 43, no. 4, pp. 8–25, 2001

[18] J. D. Sterman, *Business dynamics* : *Systems thinking and modeling for a complex world*. Boston, Irwin/McGraw-Hill, op. 2000

[19] D. Kloock-Schreiber, P. C. Gembarski and R. Lachmayer, *Application of System Dynamics for*

holistic Product-Service System Development. Procedia Manufacturing, 2020

[20] M. Hirz, W. Dietrich, A. Gfrerrer and J. Lang, *Integrated Computer-Aided Design in Automotive Development*. Berlin, Heidelberg, Springer Berlin Heidelberg, 2013

[21] J. J. Shah, Designing with Parametric CAD: Classification and comparison of construction techniques," In: KIMURA, Fumihiko (Ed.) *Geometric Modelling*. Boston, MA : Springer US, 2001, pp. 53–68

[22] S. Vajna, C. Weber, K. Zeman, P. Hehenberger, D. Gerhard and S. Wartzack, *CAx für Ingenieure : Eine praxisbezogene Einführung*. Berlin, Germany, Springer Vieweg, 2018

[23] G. La Rocca and M. J. L. van Tooren, *Knowledge-based engineering to support aircraft multidisciplinary design and optimization*. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 224, no. 9, pp. 1041–1055, 2010

[24] J. Breuker and W. van de Velde, *Commonkads library for expertise modelling : Reusable problem solving components / edited by Joost Breuker and Walter van de Velde*. Amsterdam, Oxford, IOS Press, 1994

[25] M. Stokes, *Managing engineering knowledge : MOKA: methodology for knowledge based engineering applications / edited by Melody Stokes on behalf of the MOKA Consortium*. London, Professional Engineering Publishing, 2001

[26] G. La Rocca and M. van Tooren, A Knowledge Based Engineering Approach to Support Automatic Generation of FE Models in Aircraft Design," In: *45th AIAA Aerospace Sciences Meeting and Exhibit*, 2007

[27] K. Amadori, M. Tarkian, J. Ölvander and P. Krus, *Flexible and robust CAD models for design automation*. *Advanced Engineering Informatics*, vol. 26, no. 2, pp. 180–195, 2012

[28] T. Brockmöller, H. Li, P. C. Gembarski and R. Lachmayer, *An Investigation of a Generative Parametric Design Approach for a Robust Solution Development*. *Design Conference*, pp. 315–324, 2020

[29] P. Wolniak, B. Sauthoff, D. Kloock-Schreiber and R. Lachmayer, *Automated Product Functionality And Design Optimization Instancing a Product-Service System*. *Proceedings of the Design Society: DESIGN Conference*, vol. 1, pp. 1405–1414, 2020

[30] J. Kennedy and R. Eberhart, Particle swarm optimization," In: *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, pp. 1942–1948

[31] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II*. *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002

CORRESPONDENCE

Philipp Wolniak, Research Assistant
Leibniz University Hannover
Institute of Product Development,
An der Universität 1
30823 Garbsen, Germany
wolniak@ipeg.uni-hannover.de

Daniel Kloock-Schreiber,
Research Assistant
Leibniz University Hannover
Institute of Product Development,
An der Universität 1
30823 Garbsen, Germany
schreiber@ipeg.uni-hannover.de

Bastian Sauthoff
Leibniz University Hannover
Institute of Product Development,
An der Universität 1
30823 Garbsen, Germany
sauthoff.b@gmail.com

Roland Lachmayer, Professor
Leibniz University Hannover
Institute of Product Development,
An der Universität 1
30823 Garbsen, Germany
lachmayer@ipeg.uni-hannover.de