INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.57

Procedure to Create an Automated Design Environment for Functional Assemblies

A. Osman ✉, Y. Kutay, I. Mozgova and R. Lachmayer

Leibniz Universität Hannover, Germany

✉ altun@ipeg.uni-hannover.de

Abstract

Manually exploring the solution space for different variants of a product for a given set of requirements is ineffective regarding product development time and adaptation to dynamic customer requirements. Variant generation coupled to optimization algorithms offers possibilities to search the solution space in an automated way. This paper provides a framework to build a generative parametric design environment for functional assemblies by implementing analysis as well as synthesis methods in computer-aided tools. The procedure is presented using the example of a coffee machine.

Keywords: generative design, knowledge-based engineering (KBE), design optimisation, computer-aided design (CAD)

1. Introduction

Over the last two decades Computer-aided development environments have increasingly found their way into product development. Especially in development of product variants for different use cases and requirements, they aim to partially or fully automate the design process. With increasing product complexity target group specific products are required in even shorter development cycles (Eigner et al., 2014). This results in a high number of variants, which increases the product development effort at all. Generating new conceptual changes or variants for existing products is time and cost-intensive. Even small changes to a different individuals needs can lead to high number of necessary analysis routines and model changes, which is often performed manually by design engineers with the help of adaptation and variant design methods. Challenges in the creation of automated design environments lie in the formalizability of analysis-synthesis routines as well as in building of complex product models and corresponding inference mechanisms robustly in order to avoid instability problems in case of parametric and topological changes to the geometric model (Li et al., 2019; Amadori et al., 2012). With the generation of explicit models by parametric modelling (Kim and Han, 2007), the manipulation by feature-based modelling (Yin and Ma, 2012) and knowledge-based engineering (Hirz et al., 2013; Boyle et al., 2011), several methods have already been proposed to meet these challenges. However, quick customizing of products depends on structured knowledge of design engineers, the degree of automation in design tasks as well as the usage of knowledge-based engineering methods. In this paper we discuss the use of generative design methods to realize an automated exploration of solution spaces in the concept phase of product development. Therefore after a snapshot of existing design automation methods in literature, a framework is presented to embed existing product models into automated design environments. The approach is then applied to the use case of coffee machine as a benchmark problem and the results are discussed.

2. Methods to support the design automation process

The following chapter presents some methods and tools frequently used in the literature to support partially and fully automated product design process. For this purpose, a focus is given to parametric modeling, knowledge-based engineering systems as well as generative design approaches.

2.1. Parametric modeling

In parametric modelling, properties and dependencies of product models are described by using variable quantities (parameters) in model description. The changes of parameters directly lead to changes in the model. Parameters can have a range of values or relationships among themselves, which are defined according to certain rules (arithmetic, logical or geometric dependencies). Through these rules, the product and design logic can be stored in the model (Vajna et al., 2019). This creates a generic model in which relationships between parameters are defined. Thus parametric modelling offers potentials in rapid generation of design solutions with respect to changing geometrical requirements, since the geometrical models are not static due to predefined values as in conventional CAD. This distinguishes parametric modelling from variant design (Vajna et al., 2019; Hoffmann, 2005). By changing the system parameters, not only the design structure is updated but also the geometric consistency of the model is checked (Eigner, 2014). Parametric modelling not only defines the geometry of part files, but also parametrizes at assembly level. In a CAD model, assembly files are responsible for grouping and positioning parts. Accordingly, the implementation of assembly level parameters is divided into three different categories. In the absolute positioning concept, all objects in the CAD model are defined with their positions based on the original coordinate, without creating any dependencies between geometry elements. This method is time consuming if adjustments are needed. Relative position is a multilayer dependency network and it connects objects relative to each other using a geometric constraint method. Compared to absolute positioning this procedure is a more automated method that relies on position updating. The third method is the design skeleton, which is a combination of the two methods mentioned above. The flexibility of a parametric model depends on the predefined dependencies between the parameters as well as the set of boundary conditions. Once these are set, the model loses flexibility because the solution space cannot be extended afterwards. Therefore, the design task must be studied beforehand in order to define which component variants are to be represented and which sequence of operations is necessary for this (Li et al., 2019).

2.2. Knowledge-Based Engineering Systems

Knowledge-Based Engineering Systems (KBES) can be understood as a collective term for computer aided problem solving tools (Milton, 2008). They provide knowledge-based support for the design process by reusing predefined methods, algorithms or results and integrate specific tasks or workflows involved in the design process to achieve a partially automated design of product models based on requirements and restrictions (Hirz et al., 2013; Verhagen et al., 2012). In most cases parametric modelling is used as a single tool. In order to perform an automation of the design process by means of KBES, these must have the ability to draw conclusions from existing problem descriptions and thus to eliminate the degrees of freedom in a product model by setting parameter values (Gembarski et al., 2020; Sabin et al., 1998). KBES contains domain and process knowledge (Schreiber et al., 1994). The domain knowledge represents the solution space and the process knowledge specifies the solution exploration approach by e.g. parameter constraints, formulas or design rules (Gembarski et al., 2016). For the solution space exploration and accessibility, domain knowledge need to have a formalized and structural representation. This can be done by means of rule-based, case-based or model-based reasoning mechanisms (Bellemare, 2017). Once the knowledge about the technical problem is collected and stored in a generic, knowledge-based CAD, designers can modify existing design by changing the models input specifications to quickly create and evaluate different design variants (Li et al., 2019). In general the goal of KBES is to minimize time and costs of product development by automating repetitive, non-creative design tasks, supporting multidisciplinary design optimization at all stages of the design process and thus providing support for solution space exploration (La Rocca, 2007). One well-known methodology to support a structured development of KBES is MOKA (Methodology and Tools Oriented

https://doi.org/10.1017/pds.2022.57 Published online by Cambridge University Press

to KBE Applications). MOKA describes the capturing of engineering knowledge and the integration of it into KBE applications in six phases (Stokes, 2001):

- Identify: Determination of objectives and assessing technical feasibility.
- Justify: Evaluating commercial and technical aspects.
- Capture: Collecting and structuring of knowledge in informal models.
- Formalize: Transforming structured data into product/process models
- Package: Converting formalized models into code to be used in KBE.
- Activate: Distribution and installation.

2.3. Generative Design Approaches

Generative modelling is a method in which a components shape is iteratively adapted using existing input data. For this purpose rules from existing engineering knowledge are used to autonomously generate a high number of alternative solutions that meet specific requirements as well as restrictions ([Li et al., 2019; Sabin et al., 1998). Generative modelling usually starts by expressing design ideas through a set of algorithmic rules. Generative inference mechanisms allow design solutions to be automatically generated and visualized as outputs (Schreiber et al., 1994). Most of the generative design approaches uses graph grammar and spatial grammar to generate CAD Models. These have the potential to create structural components that can be used directly in a CAD environment (Gembarski et al., 2016). Graph grammar has a vocabulary of graphs and a set of rules to build a CAD model through a network of graphs. The set of rules has mechanisms that can identify structures within the graph network and replace them with another (Li et al., 2019; Cui and Tang, 2017). The spatial grammar uses a geometry representation and a vocabulary consisting of geometric parameters instead of graph network (Krish, 2011). Currently, it is still a challenge to formalize engineering knowledge via rules. This is due to the large number of geometric dependencies and the geometric complexity in case of many sub components. CAD models can be represented by engineering languages, but the questions of how to many rules are needed and how to develop these rules and vocabulary based on existing knowledge need still to be investigated (Chakrabarti et al., 2011). According to Kris et al. the generative design process can be categorized in three main elements: A design scheme, a means of creating variations and a means of selecting desirable outcomes. (Krish, 2011). In addition, deep learning methods are increasingly being used for generative design as well. Oh et. al describe the application of generative adversarial networks for generating different design solutions with regard to the optimization of topologies (Oh et al., 2019).

3. Generative Parametric Design Approach (GPDA)

GPDA has been introduced for optimization of structural components (Li et al, 2019). To achieve efficient design solutions, GPDA combines the generative aspects of automatically replacing parts of a product model with the advantages of parametric modelling and thus leads to dynamic and robust product models. The basic idea of GPDA is to reduce global dependencies in CAD models, as these limits the flexibility. This can be accomplished by separating the model structure from the product structure of a component or assembly. Thus individual components no longer have to be a single CAD model but rather can be combination of several. Regarding assemblies the interfaces of the individual components are no longer where they are located in a physically existing product, instead they are at those points where the lowest dependencies for the overall model occur. The areas between interfaces are called design zones. These serve as placeholders for design elements that represents the individual solution variants of the design zone. Crucial in the design zone is the definition of reference interfaces, where reference geometry in the form of planes or curved surfaces is defined using a skeleton within the CAD model. The reference interfaces connect two design zones and ensure a continuous geometry transition. In addition to these interfaces, the dimensions and arrangements of the design zones are also defined by global parameters. The skeleton thus represents the highest level in the dependence chain, which is why its geometric stability must be guaranteed in any case. Within a design zone, design elements can be exchanged. This allows the GPDA to cover a larger solution space compared to purely parametric modelling, since topological changes can be made to the product shape in addition to scaling adjustments. (Li et al., 2019; Herrmann et al., 2021). In order to automatically generate an optimized

https://doi.org/10.1017/pds.2022.57 Published online by Cambridge University Press

geometry through the above described model structure, a computer-aided development environment is required. This adapts the component geometry iteratively by alternating synthesis and analysis steps until the requirements are met, e.g. with an evolutionary algorithm. As design variables of the optimization algorithm the parameters of the design zones or design elements are used. As objective functions can requirements as well as certain user-based characteristics of the product can be implemented. In a previous work we introduced an overall architecture for GPDA environments (Figure 1) in order to adapt the idea of GPDA to other domains besides structural engineering (e.g. functional assemblies) (Herrmann et al., 2021).

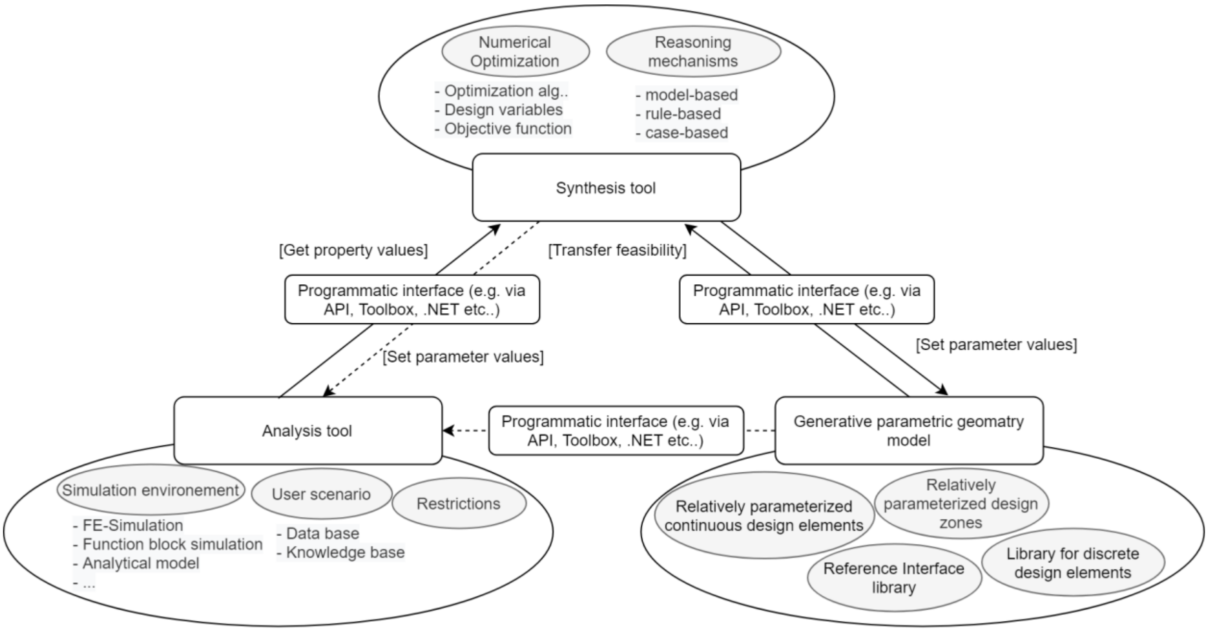

Figure 1. Design environments according to the GPDA

4. Problem analysis and scope of research

The need for methods and development environments for automated design of products and variants is still a research topic of many scientific publications (Holder et al., 2019; Bursac, 2016; Schleich et al., 2015). Generative design environments offer great potentials with regard to the possibility of virtual testing of different variants for given requirements before introduction to the market. Being able to compare and evaluate different variants and concepts for given usage scenarios with computer-aided tools accelerates the design process and increases its sustainability. For the integration of computer-aided tools in the structure of such design environments, a representation of analysis and synthesis methods is necessary. In particular the implementation of synthesis methods is a challenge regarding the formalization. Approaches like graph-based languages go as far as to implement geometric properties into formalized modeling languages like UML or SysML and build-up a complex network of dependencies. These modeling languages either mainly focus on the product structure or only abstractly link sub-assemblies or parts with actual functional outcome or they are based on a discrete network which is rather inflexible upon changes. GPDA is a promising concept, because it offers many advantages to design engineers compared to other modelling approaches mentioned in section 2., which are listed below: (Li et al. 2019, Wolniak et al., 2020)

- Reduction of interdependencies by distinguishing modeling structure from assembly structure.
- Design elements are themselves parametric and can be developed independently.
- More robust solutions by establishing the relationship between functional and physical representations.
- Reusability of individual design elements stored in libraries for new product usage scenarios.
- Enlargement of the solution space by adding new design elements
- Automated comparability of conceptual changes.

While the effort for setting up a GPDA environment is higher compared to manual or knowledge-based design, the design effort for a new variant is lower in comparison (Figure 2) (Hermann et al. , 2021). In manually design process, a few variants are evaluated to find a solution. In knowledge-based design, a higher number of variants can be generated semi-automatically by means of reasoning mechanisms. Coupling generative parametric models to numerical optimizers leads to a further solution space restrictions, but offers the potential to vary parameters of a geometric model as design variables in smaller steps and thus to evaluate more variants and concepts.

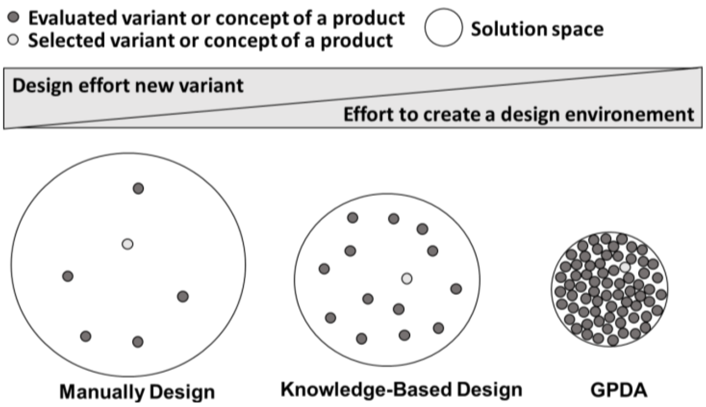

Figure 2. Evaluation of concepts within a solution space

In this work we investigate the transfer of GPDA to assembly level with a focus on functional assemblies. Functional assemblies are defined here as technical systems where the mass and energy flow between sub components of an assembly plays a significant role to satisfy the customer needs. The goal is to provide a framework that supports design engineers in creation of automated design environments for such assemblies and thus to help design engineers to identify the optimum for a variant/concept of a product faster than through manual iterations. Therefore engineers could apply the framework to their product portfolio and can react to new requirements through automated customization. The research questions to be answered here are following one:

How to create a design environment, which automatically optimizes functional assemblies with respect to defined use cases and target functions?

How can the knowledge transfer between a simulation domain, a geometrical domain and an optimization domain be realized?

5. Procedure to create a GPDA environment for automated exploration of the solution space

In order to transfer the GPDA concept to a designers own product model, the architecture of a GPDA design environment presented in chapter 3 (Figure 1) must be supplemented by a methodical approach broken down into individual steps. This section presents an approach to support the designer during the development of the GPDA environment and focuses on functional assemblies as defined in chapter 4. Therefore for the analysis domain a numerical optimization tool and for the synthesis domain a functional block simulation has been chosen. As already mentioned, the approach proposed here focuses on the concept phase of product development. It can therefore be assumed that the product architecture is known as well as partial solutions for particular functions can be captured morphologically. Additional to the GPDA components mentioned in chapter 3 it is needed to distinguish the design elements of an assembly as following:

- ***Continuous design elements***: A continuous design element is parameterized both within itself and relative to the associated design zone. These are components of the assembly that are variable in their geometric dimensions and can be alternated through optimization algorithms.
- ***Discrete design elements***: Discrete design elements are components that are parameterized only in their positions relative to the design zone. They are standard components or purchasing components and stored in libraries (databases) to be alternated through optimization algorithms.

https://doi.org/10.1017/pds.2022.57 Published online by Cambridge University Press

The procedure for developing the GPDA development environment is shown in Figure 3.

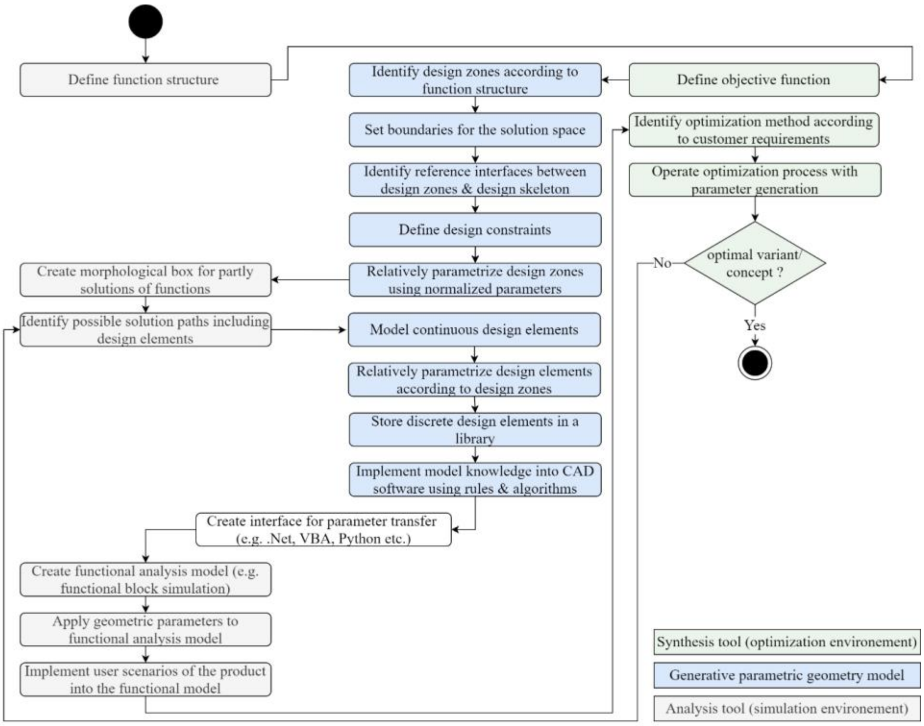

Figure 3. Procedure to create a GPDA environment for functional assemblies

6. GPDA environment of a coffee machine

6.1. Generative parametric geometry model

To illustrate the idea of GPDA for functional assemblies this chapter focuses on an application example of a coffee machine. Based on the analysis of the products function structure, eleven design zones are defined, which are shown in Figure 4(a). Each design zones contains design elements (1 element to n elements). For example the design zone drawer contains some discrete design elements pump, heating element and elements for the transport of water/coffee such as tubes and hoses. Design zone brewing chamber consists of two continues design elements, which are the brewing chamber (Figure 4 (c), yellow component) for coffee itself and a surrounded one for waste storage (Figure 4 (c) red component).

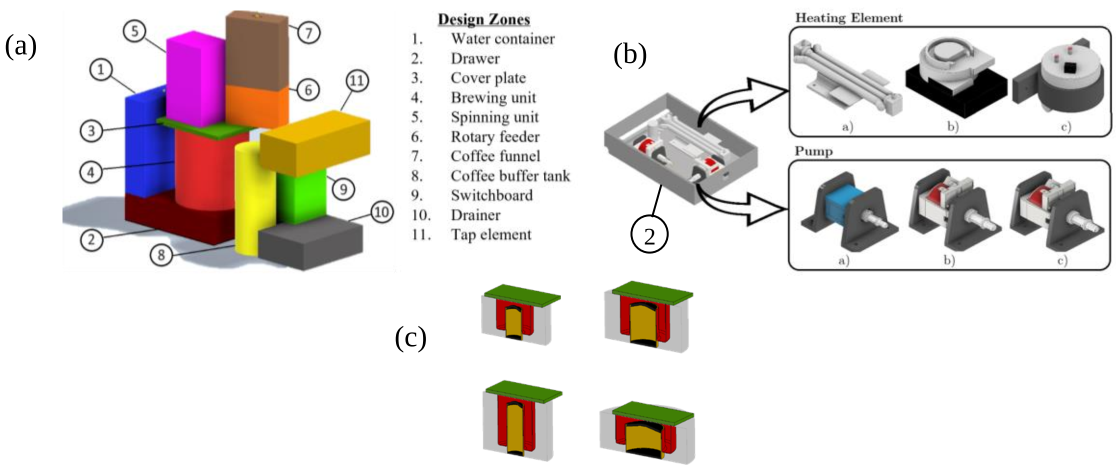

Figure 4. Geometry domain of the GPDA - (a) Design zones of the coffee machine; (b) Example of variants of discrete design elements within the design zone drawer; (c) Example of variants of the continuous design element brewing chamber

https://doi.org/10.1017/pds.2022.57 Published online by Cambridge University Press

Once the design zones are identified and design elements are assigned, parameterization begins. Therefore the overall boundaries of the solution space has to been defined (maximum size of the coffee machine). In a next step, the design zones are parameterized relative to each other using the design skeleton in three axis (Figure 5). The skeleton is represented through the reference interfaces, which will be drawn on the planes of the skeleton depending on the design elements located at the interface.

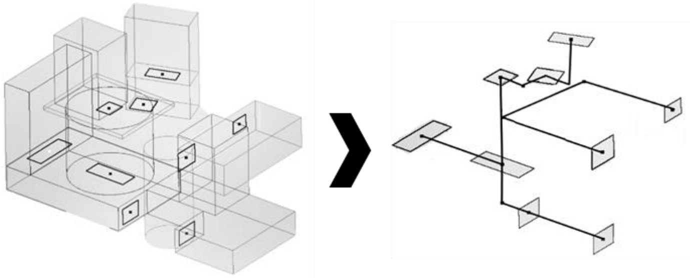

Figure 5. Design skeleton of the coffee machine

Afterwards the design elements are described with the parameters of the design zones. I.e. for the design zones the design skeleton provides the leading parameters, for the design elements the design zone provides the leading parameters. The design elements are parameterized both in itself and in relation to their position within the associated design zone. Once the parametrization is done, the logic is implemented rule-based in the CAD software. In order to be able to compare different solutions for a design element in an optimization environment, the desired variants for the design elements are stored in libraries (e.g. different types of pumps shown in Figure 4). The individual design elements are rule-based assigned to values in the CAD Software which are later used as design variables in the optimization environment. This is how the exchange of design elements can be realized.

6.2. Functional simulation model and optimization environment

The next step is to set up a simulation model in which the essential functions of the coffee machine can be simulated. For this purpose, a function block simulation is set up in Matlab/Simulink. Here, the performance of the coffee machine is expressed in functions with the help of the geometric parameters of the GPDA model (e.g. diameter of brewing chamber) and the performance values of the functional components such as pumps and heating elements. The functional model must be able to calculate objective-function relevant variables. In the case of the coffee machine, variables such as operating time for a certain number of cups and water consumption could be used. Depending on the size and variant of the design elements, these customer-relevant quantities will change. Furthermore, the implementation of a user scenario in the function block simulation is required in order to be able to represent the time-based consumption behavior at the coffee machine. For this purpose, look-up tables can be used in Simulink such as shown in Figure 6.

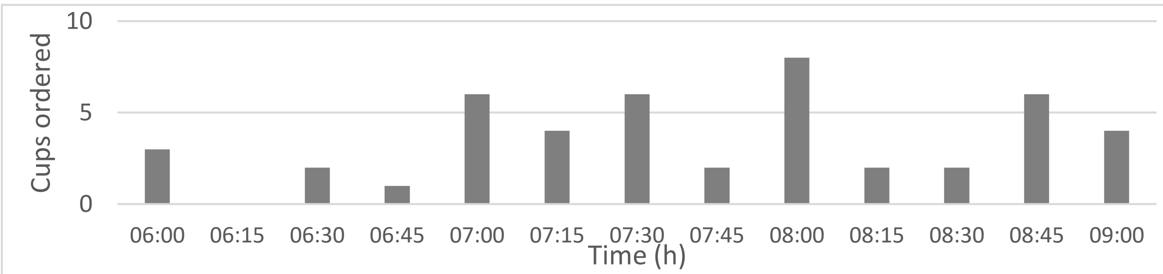

Figure 6. User-scenario in office environment

Once the function block simulation is set up, the output of the simulation has to be coupled with an optimization algorithm. For example, the water consumption and the operating time are used as objective functions in a multi-objective optimization environment. In both cases a minimum is the objective. The geometric parameters of the design zones, which are relevant with regard to the objective function are used as design variables of the optimization algorithm. Furthermore, the use of different

https://doi.org/10.1017/pds.2022.57 Published online by Cambridge University Press

solutions of design elements are also varied via design variables. Figure 7 shows the optimization process. In Autodesk Inventor, the parameters of the design zones are described as user parameters. Relative to these, the design elements are parameterized as described above and also saved as user parameters. The design zone parameters are alternated via the optimization algorithm (top right of Figure 7). These x values are alternated between 0 and 1, which leads to increasing and decreasing of the design zones. This in turn leads to specific design element dimensions, which are transferred back to Matlab/Simulink via the .Net interface. With the help of these values (e.g. diameter of the design element scalding chamber), the objective function-relevant variables are calculated (e.g. number of available coffee at a defined time period). Figure 7 further shows different expressions of the design zones during optimization. The "drawer" and "coffee buffer tank" design zones are also shown as examples with the different variants of the design elements during optimization.

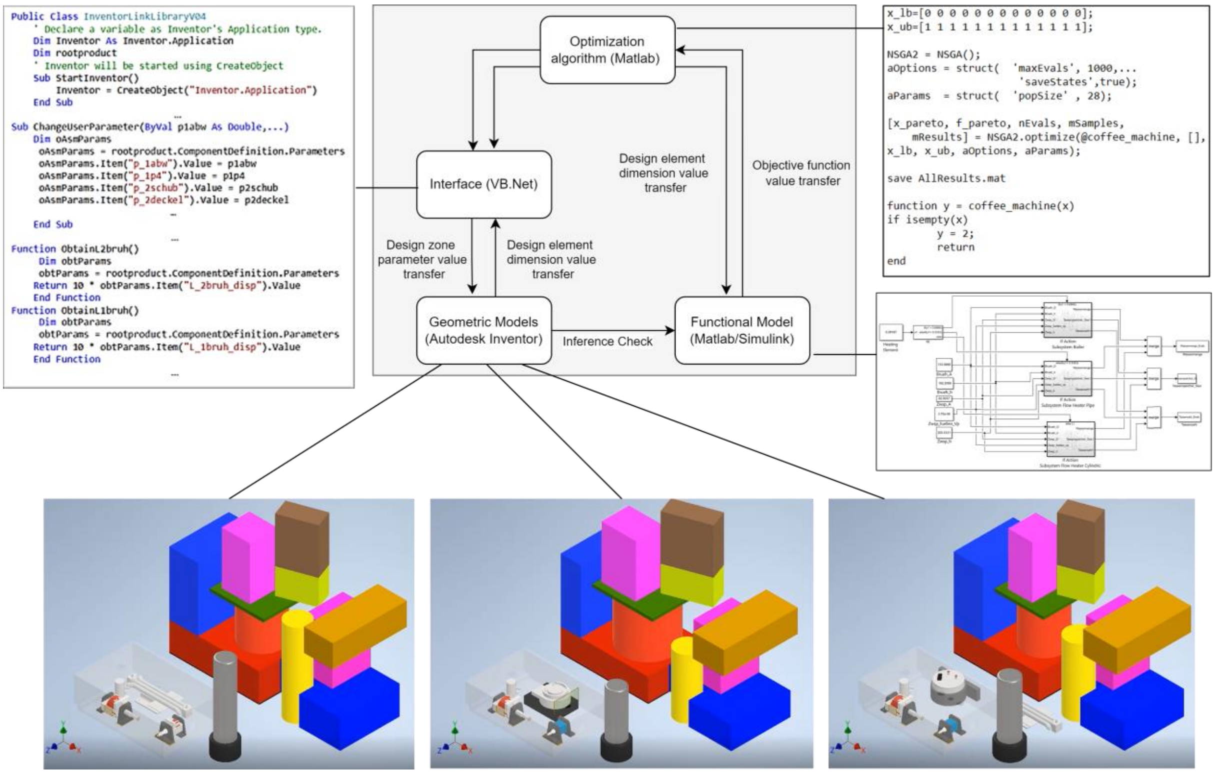

Figure 7. Optimization process and parameter transfer between GPDA domains

6.3. Results of the multi-objective optimization

For the optimization the non-dominated sorting genetic algorithm (NSGA-II) was used, which is a multi-objective approach and performs reasonably in design problems (Liddicoat, 2016). The selected user scenario is a period of three hours in the morning where guest have to be served at several peak hours with a cup of coffee (Figure 6). As design variables normalized parameters of the design zones (top right of Figure 7) have been used as well as different types of heating elements and pumps. The alternation of the different pumps and heating elements in the simulation was enabled through assigned values between 0 and 1. The results of the multi-objective optimization are plotted in Table 1 and Figure 8. Each point in Figure 8 refers to different settings of the design variables and thus represents different geometric values of the product model in the GPDA model. For example, Figure 8 shows the values of the design variables by which pump and heater types are assigned in the geometric model. Also, each value of the water consumption-working time combination can be assigned geometric values of the design elements in the GPDA model. Thus, each point in Figure 8 represents a variant. The optimal variants are located on the Pareto front. Based on this information, it was determined that the accumulation of solution points on the right represents small pump scenarios. There is also another vertically clustered group on the far right of the graph. Here are the scenarios which belong to the beyond

https://doi.org/10.1017/pds.2022.57 Published online by Cambridge University Press

of the specified simulation time. On the other hand, according to the multi-objective analysis, it can be seen that the pareto front points (the far-left cluster) that dominate the other solution points are provided by using the concept of flow heater (pipe). Both Figure 4 and Figure 7 show some different geometric representations during optimization.

Working hours	Water Consumption	Pump Parameter	Heating Element Parameter	Pump type	Heating element type
2,720	29,833	0,907	0,281	Big	Flow heater (Pipe)
2,910	26,492	1	0,041	Big	Flow heater (Pipe)
3,433	24,971	0,572	0,291	Medium	Flow heater (Pipe)
3,489	26,093	0,798	0,952	Big	Boiler
5,475	24,318	0,889	0,385	Big	Flow heater (Cylindric)
5,815	23,573	0,451	0,362	Medium	Flow heater (Cylindric)
6,012	24,642	0,428	0,417	Medium	Flow heater (Cylindric)
8,217	27,324	0,265	0,145	Small	Flow heater (Pipe)
9,105	24,173	0,274	0,842	Small	Boiler
9,722	21,498	0,233	0,565	Small	Flow heater (Cylindric)
9,722	20,533	0,088	0,398	Small	Flow heater (Cylindric)

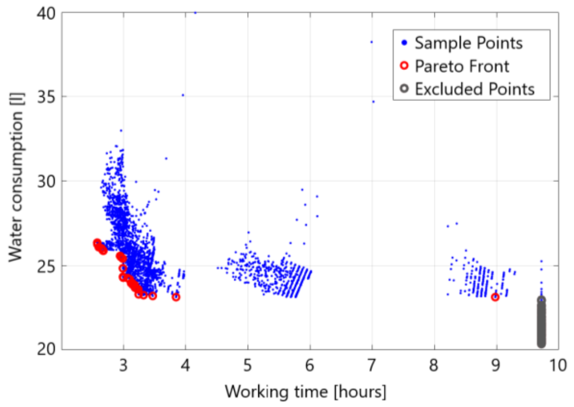

Figure 8. Objective value space

7. Conclusion and Outlook

In the context of this work, the applicability of generative design approaches, in particular GPDA, to functional assemblies has been investigated. After a basic motivation of generative design environments and their potential in rapid variant generation and mass customization, the GPDA approach for structural components has been transferred to functional assemblies. The necessary methods and tools have been discussed and recorded in a framework. The proposed generative development environment consists of three domains: an analysis tool, a synthesis tool and a generative parametric geometry model. The linking of a geometry model, an optimization and a simulation environment can enable the automatic search for optimal variants within a restricted solution space of a product. A methodical approach has been proposed for the application of such design environments in the concept phase of product development. The developed approach was discussed using an example of a coffee machine. The results show that with the help of a GPDA environment, which uses numerical optimization as a synthesis tool and parameters of a generative parametric geometry model as design variables, optimal variants can be identified for given user scenarios. The main advantage of this fully model-based design environment is that the libraries for design elements can be extended at any time and existing solutions can be reused for new requirements. The effort required to develop such GPDA environments may be high, but at the same time it offers the potential to react quickly to dynamic changes in the market. The developed framework needs to be investigated in future work on further application examples to ensure its generalizability. Future challenges are to guarantee the relative parameterization of moving parts within the GPDA when these parts move across several design zones and providing a methodical framework to identify optimal design zones.

References

Amadori, K. et al. (2012), “Flexible and robust CAD models for design automation”, *Advanced Engineering Informatics*, Vol. 26 No. 1, pp. 180-195. https://doi.org/10.1016/j.aei.2012.01.004

Bellemare, J.; Carrier, S.; Nielsen, K.; Piller, F. T. (2017), "Managing Complexity", *Springer International Publishing*, Cham.

Boyle, I., Rong, Y. and Brown, D. (2011), “A review and analysis of current computer aided fixture design approaches”, *Robotics and Computer-Integrated Manufacturing*, Vol. 27 No. 1, pp. 1-12. https://doi.org/10.1016/j.rcim.2010.05.008

Bursać, N. (2016), *Model Based Systems Engineering as a support for the Modular Design in the Context of the Early Stages of Product Generation Engineering*, Karlsruhe, PhD-Thesis, 2016.

https://doi.org/10.1017/pds.2022.57 Published online by Cambridge University Press

Chakrabarti, A. et al. (2011), “Computer-based design synthesis research: an overview”, *Journal of Computing and Information Science in Engineering*, Vol. 11 No. 2, p. 021003. https://doi.org/10.1115/1.3593409
Cui, J. and Tang, M. (2017), “Towards generative systems for supporting product design”, *International Journal of Design Engineering*, Vol. 7 No. 1, pp. 1-16. https://doi.org/10.1504/IJDE.2017.085639
Eigner, M., Roubanov, D., Zafirov, R., *Modellbasierte virtuelle Produktentwicklung*. 1. Aufl. Berlin, Heidelberg: Springer-Verlag, 2014.
Gembarski, P.C. , Bibani, Mehdi, and Roland Lachmayer (2016) “Design Catalogues: Knowledge Repositories for Knowledge-Based-Engineering Applications.”, in Marjanović, Dorian, Štorga, Mario, Pavković, Neven, Bojčetić, Nenad and Stanko Škec (eds) *Proceedings of the DESIGN 2016 14th International Design Conference*, Glasgow, The Design Society
Gembarski, P.C. (2020), "On the conception of a Multi-Agent Analysis and Optimization Tool for Mechanical Engineering Parts". *Agents and Multi-Agent Systems: Technologies and Applications, Smart Innovation, Systems and Technologies*, vol. 186.2020, pp. 93-102, 2020. https://dx.doi.org/10.1007/978-981-15-5764-4_9
Herrmann, K., Altun, O., Wolniak, P., Mozgova, I., Lachmayer, R., " Methodischer Aufbau von Entwicklungsumgebungen nach dem Generative Parametric Design Approach." *Proceedings of the 32nd Symposium Design for X (DFX2021).* https://doi.org/10.35199/dfx2021.14
Hirz, M. et al. (2013), *Integrated computer-aided design in automotive development*, Springer, Graz. https://doi.org/10.1007/978-3-642-11940-8
Hoffmann, C. (2005), “Constraint-based computer-aided design”, *Journal of Computing and Information Science in Engineering*, Vol. 5 No. 3, pp. 182-187. https://https://dx.doi.org/10.1115/1.1979508
Holder, Kevin; Rudolph, Stephan; Stetter, Ralf; et al: Automated requirements-driven design synthesis of gearboxes with graph-based design languages using state of the art tools. Forsch Ingenieurwes 83, 655–668, 2019. https://doi.org/10.1007/s10010-019-00322-z.
Kim, B. and Han, S. (2007), “Integration of history-based parametric translators using the automation APIs”, *International Journal of Product Lifecycle Management*, Vol. 2 No. 1, pp. 18-29. https://doi.org/10.1504/IJPLM.2007.012872
Krish, S. (2011), “A practical generative design method”, Computer Aided Design, Vol. 43 No. 1, pp. 88-100. https://doi.org/10.1016/j.cad.2010.09.009
La Rocca, G., Van Tooren, M., "A Knowledge Based Engineering Approach to Support Automatic Generation of FE Models in Aircraft Design,*" In: 45th AIAA Aerospace Sciences Meeting and Exhibit*, 2007
Li, H. and Lachmayer, R. (2019), “Automated Exploration of Design Solution Space Applying the Generative Design Approach”, *Proceedings of the Design Society: International Conference on Engineering Design, Cambridge University Press*, Vol. 1, No. 1, pp. 1085-1094. https://doi.org/10.1017/dsi.2019.114
Liddicoat, D. E.: *An Automated Interface for CATIA and MATLAB with an Op-timisation Capability*. University of New South Wales at the Australian De-fence Force Academy; 2016.
Milton, Nick.R. (2008), *Knowledge technologies*. Monza, Polimetrica sas.
Oh, S., Jung, Y., Kim, S., Lee, I., and Kang, N. (2019). "Deep Generative Design: Integration of Topology Optimization and Generative Models." ASME. *J. Mech. Des*. November 2019; 141(11): 111405. https://doi.org/10.1115/1.4044229
Sabin, Daniel, and Rainer Weigel (1998) “Product configuration frameworks - a survey*.” IEEE intelligent systems* 13(4): 42-49.
Schreiber, G., Wielinga, B., De Hoog, R., Akkermans, H., Van de Velde, W. (1994) “CommonKADS: A comprehensive methodology for KBS development.” *IEEE expert* 9(6): 28-37.
Schleich, B., Wartzack, S., "A generic approach to sensitivity analysis in geometric variations management." *DS 80-4 Proceedings of the 20th International Conference on Engineering Design* (ICED 15 Vol 4: Design for X, Design to X, Milan, Italy, 27-30.07, 2015.
Stokes, M. (2001), *Managing Engineering Knowledge - MOKA: Methodology for Knowledge Based Engineering Applications*, Professional Engineering Publishing Limited, London.
Vajna, S., Weber, C., Zeman, K., Hehenberger, P., Gerhard, D., Wartzack, S. (2018), *CAx für Ingenieure – Eine praxisbezogene Einführung*. 3. Aufl. Berlin: Springer-Verlag, 2018.
Verhagen, Wim .J.C., Bermell-Garcia, Pablo, van Dijk, Reinier E.C., and Richard Curran (2012) “A critical review of Knowledge-Based Engineering: An identification of research challenges.” *Advanced Engineering Informatics* 26(1): 5-15.
Wolniak, P.; Klookschreiber, D.; Sauthoff, B.; Lachmayer, R., "Integrating Architectural Design Changes in Computer-Aided Design Optimization." *Internation Conference on Mass Customization and Personalization - Community of Europe* (MCP-CE 2020).
Yin, C. and Ma, Y. (2012), “Parametric feature constraint modeling and mapping in product development”, Advanced Engineering Informatics, Vol. 26 No. 3, pp. 539-552. https://doi.org/10.1016/j.aei.2012.02.010