10th International Conference on Mass Customization and Personalization – Community of Europe (MCP - CE 2022)
Toward the Sustainable, User-Centric and Smart Industry 5.0
September 21-23, 2022, Novi Sad, Serbia

USING AXIOMATIC DESIGN FOR THE DEVELOPMENT OF PRODUCT CONFIGURATION SYSTEMS

Marcus Oel, Paul Christoph Gembarski, Roland Lachmayer
Institue of Produkt Development, Leibniz University Hannover,
An der Universität 1, 30823 Garbsen

Abstract: *In order to meet a wide range of customer requirements in product development, a high degree of individualization is necessary today, which can be achieved with product configuration systems. Knowledge-based CAD models are a useful tool for implementing such configurators, but they are significantly more complex to develop than conventional parametric CAD models. To master this complexity, this article examines the use of the axiomatic design approach for the development of a configurator for a skip loader. In combination with the parameter space matrix, an application-oriented methodology is derived which is suitable for the development of similar configuration systems.*

Key Words: ***Axiomatic Design, Product Configuration, Parameter Handling, Parameter Space Matrix***

1. INTRODUCTION

As the trend towards mass customization continues to be an important aspect of today's product development and since there is also high time and cost pressure, it makes sense to use configurators in the product development process [1]. In combination with a CAD system and a calculation environment, not only a pure product configurator can be developed, but also the construction and design of products according to customer requirements can be realized in the form of a knowledge-based system (KBS) [2]. Overall, a significant time saving in the development cycle can be achieved by using a KBS, but the scope of tasks changes from routine activities to complex and creative activities to cope with the more complex development task [3]. An important tool in product development is the use of appropriate methodologies, which enable structured and targeted problem solving [4]. Methodologies for the development of KBS, like e.g. CommonKADS or MOKA, have been discussed as solution centered approaches that enrich design models with relevant knowledge artefacts [5, 6].

In product development, many methodologies target on decomposing a design problem into sub-problems and solve them in a structured manner. Some design methodologies also link knowledge development and documentation, one of them is axiomatic design [7]. The core of this methodology are the two axioms: The independence axiom and the imformation axiom. The goal of applying these axioms is the structuring of complex development tasks. In the context of KBS it can be used to streamline used parameters. In order to achieve this, unnecessary dependencies is to be avoided (independence axiom) and the total information content is to be reduced (information axiom). Since the guidelines of such a methodology can limit product development in addition to the benefits, this article will examine the application of axiomatic design in the construction of a product configurator. The aim is to investigate the suitability of axiomatic design for the application-oriented implementation of a KBS.

In Section 2 the basics of the KBS will first be discussed and suitable methods as well as axiomatic design in detail and the parameter space matrix will be presented. Subsequently, in section 3 a product configurator for skip loaders is developed and, based on the findings from the application of the presented methods, an adapted process model is developed before the results are subsequently evaluated and discussed in section 4.

2. RESEARCH BACKGROUND

2.1. Configurators as Knowledge Based System

Knowledge-based systems (KBS), also known as expert systems, can solve tasks using artificial intelligence methods. The procedure for solving problems is modeled on that of a human expert and is especially suited to solve complex tasks [8, 9]. Furthermore, with an appropriate knowledge base it is possible to solve interdisciplinary problems. This changes the task area of the involved developer from routine activities to creative tasks, whereby overall the necessary time can be minimized and thus personnel can be saved [3].

In combination with a CAD system, this results in a powerful tool for a wide range of engineering problems, which is also referred to as knowledge-based engineering

(KBE) systems. Product configuration systems that rely on CAD functionalities for visualization purposes and further processing of geometric data are an instance of KBE systems [9]. Compared to traditional approaches of parametric modeling in today's CAD systems, KBE shifts the focus from designing a single variant to developing solution spaces [10].

Besides the actual knowledge base, a necessary part of KBE systems is the inference engine that allows the model to draw conclusions from the design context [11, 12]. Based upon different concepts like, e.g., production rules or a model representation via domains and constraints, the KBE system is able to automatically reason about the fit for a product variant for changed requirements and execute necessary changes [13].

2.2. Methodologies for KBE System Development

A large number of methodologies exist for the general development of technical products, such as guidelines like VDI-2221 [4]. For the development of KBE systems, there exists a manifold of specialized methods that provide procedures and models to KBE-system-specific challenges, like e.g. knowledge acquisition and formal modeling [9].

CommonKADS, MOKA, MIKE, KNOMAD and KAMET II are examples of these methodologies [5, 6, 14–16]. These all differ somehow in their objectives and focus. MIKE, KNOMAD and KAMET II focus primarily on the use of proprietary tools for selected steps in the development cycle of a KBE system. In comparison, CommonKADS focuses on the organizational level for structuring the necessary work steps and formal modeling of knowledge, stakeholder roles and the later design artifact [6]. MOKA provides a six-step process model for KBE system development, the MOKA lifecycle. It primarily supports knowledge engineering, i.e., the acquisition, formalization and implementation of available knowledge into design artefacts. Thereby, the focus is on knowledge management and informal modeling [7]. Due to their more general nature, CommonKADS and MOKA have reached dissemination.

To a certain extent, KBE system development implies a solution oriented approach to modeling. Usually, there is some kind of existing design or model that can be equiped with knowledge and reasoning. A different approach is algorithm-aided design and computational design synthesis which is to be understood as problem-oriented approach [17]. The aim is to capture the laws of creation how a design artifact is developed and to create a draft generator on this basis [18, 19]. Applications are, e.g., additively manufactured nozzles [20] or patient specific bone-anchored implants [21].

2.3. Axiomatic Design

The axiomatic Design approach is not so much aimed at the processes of knowledge implementation, but rather provides a methodology for structured problem solving in a complex system [8]. The focus is primarily on the development of the product and process parameters starting from the initial needs. In detail, the development process is divided into four domains. The customer domain contains the general customer needs. Subsequently, the functional domain is used to formulate the functional reqirements, which are similar to an ordinary requirements list [22]. The third section, the physical domain, contains all the design parameters of the product to be developed, while in the fourth domain process variables are recorded in the process domain. The dependencies between the parameters of a domain and those of the subsequent domain can be described by a design matrix. This is multiplied by the corresponding parameter vectors [11].

In addition, axiomatic Design provides two principles, which are referred to as axioms. The first axiom, called Independence axiom, requires an independence of the functional requirements. The Information axiom, on the other hand, requires that the information content of the system be minimized [23].

2.4. Parameter Space Matrix

The parameter space matrix (ParSM) presented in [24] is intended to simplify the handling of parameters in CAD design in connection with requirements and restrictions. For this purpose, the parameters of a component are derived from requirements applicable to it. At the same time, the parameters are linked to any restrictions. Thus, on the one hand, the dependencies between requirements and parameters are clearly visible and, on the other hand, violations of the restrictions quickly become apparent. By creating the ParSM in an Excel spreadsheet embedded in the CAD system, the parameters can be used directly for the design [24, 25]. The use of an Excel spreadsheet offers the further advantage that a variety of operators are available for calculations and solving operations.

3. DEVELOPMENT OF A PRODUCT CONFIGURATOR USING AXIOMATIC DESIGN

In this article, the practical application of the axiomatic design methodology in combination with a matrix-based parameter management, the by Gembarski [24] developed ParSM, is investigated. The goal is to achieve a structuring of the parameters and their dependencies and at the same time to enable an easy and user-friendly implementation into a CAD program. For this purpose, a configurator for the superstructure of skip loaders is developed and the specifications of the methodology listed above are applied to this KBE system.

Trucks that can pick up, set down and empty containers of different sizes are referred to as multi-bucket system vehicles according to DIN 70723-1 [26]. Another common designation is skip loader, which will be used in this article. The actual superstructure, which performs the function of picking up, setting down and emptying the container, is mounted onto the vehicle frame of a truck and is referred to as a skip loader. Manufacturers of such superstructures are usually not the actual vehicle manufacturers, but application-specialized companies for vehicle construction. This means that customized solutions are necessary, since in addition to different application requirements, the connection to different basic vehicles must also be made possible. The focus of the product configurator presented in this article is therefore not on developing a solution principle for a skip loader.

For such a problem, design catalogs can be used, as shown by Roth [27]. Rather, a generally working solution is to be developed whose solution space includes many variants. The decisive factors for the solution space in this case are the size of the container and the available payload or vehicle weight. The bandwidth examined here ranges from 2m³ to 20m³ and vehicle weights from 7.5to to over 30to.

The CAD system Autodesk Inventor in combination with Microsoft Excel is used to implement the configurator as a KBE system. In Fig. *1* the information flow within and between the Software components is shown.

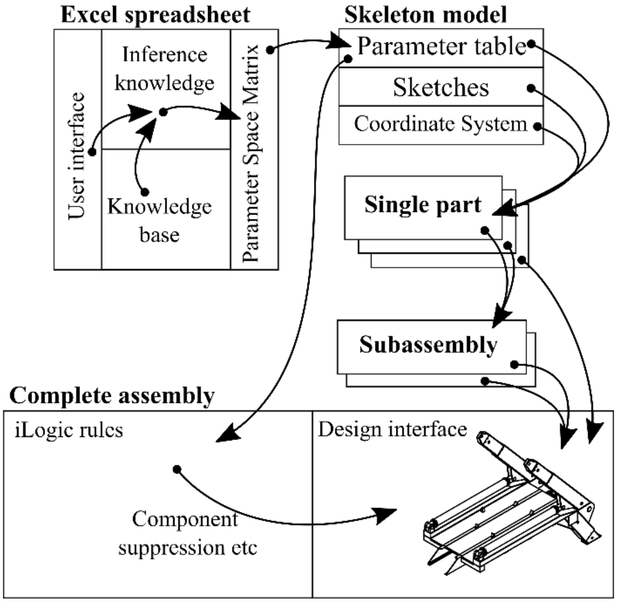

Fig. 1. *Information sharing within the KBS*

The user's input through the user interface is further processed within the Excel spreadsheet by using the embedded knowledge. The result of this are the parameters necessary for the construction. These are retrieved from a skeleton model from which the individual parts are derived, as described by Gembarski [25]. In this KBS, the skeleton model represents the link between the knowledge and engineering systems. The placement of all individual parts and subassemblies is carried out by means of a coordinate system created in the skeleton model. In addition, actions such as the suppression of individual components are solved by means of Autodesk Inventor's own programming environment, iLogic.

The final model of the skip loader is shown in Fig. 2. The design is based on a subframe that can be individually adapted to a carrier vehicle and adapts to the configured overall size of the body. This consists of a welded construction of standardized structural hollow sections. There are vehicle manufacturer-specific guidelines that must be observed. These include, for example, the angulation of the main longitudinal beam in the front area or the stiffening of the frame by closing the profile in the rear area, as shown in Figure 3.

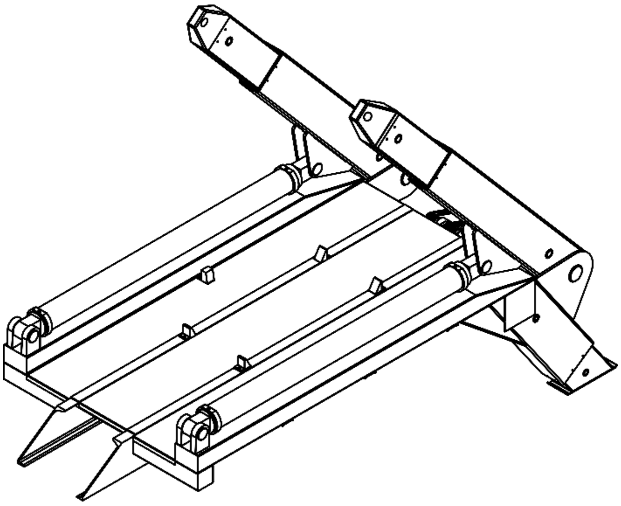

Fig. 2. *Design of the skip loader*

The main function of setting down a container is performed by two swivel arms, which also have a telescopic function. The emptying of a container is done by one to three hooks, depending on the type of container. Stability is ensured by a telescopic support. Hydraulic cylinders with a standardized sizing are used as drives for the above functions. The hydraulic components of the telescopic function of the swivel arm and the support as well as the design of the hook for emptying are shown by the breakouts in Fig. *4*.

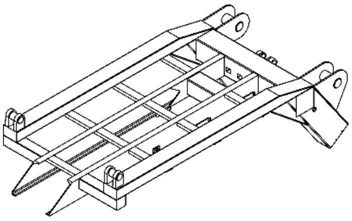

Figure 3: Subframe

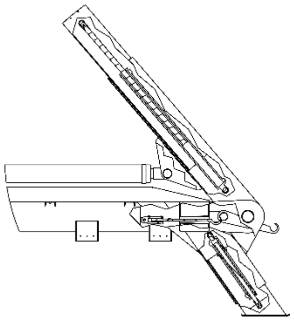

Fig. 4. *Telescopic function*

The range of functions of the KBS itself includes a large number of calculations. First, the geometric kinematics of the swing arms are established. For this, the coordinates of the specific points from Fig. *5* must be determined. This is not done by sketches within the CAD environment, but by geometric calculations within the Excel spreadsheet to avoid a backtracking of the information from CAD to Excel. On this basis, all structural hollow sections are dimensioned. For this purpose, strength calculations are carried out for the most heavily loaded parts, which are the swivel arms and the telescopic support. In addition, a weld seam calculation is carried out. This is done exemplarily for a characteristic point, which is the connection of the swivel arms to the bearing plate of the hydraulic cylinder for the swivel movement. Finally, the stability is checked. For this purpose, the mass and center of gravity of the finished design of the superstructure and the carrier vehicle are used.

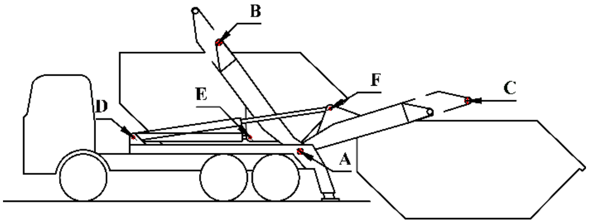

Fig. 5. *Points of the swivel arm kinematics*

3.1. Defining Domains and their targets

The methodology of axiomatic Design according to Suh [28] was used for the structure of the KBS. The basic structure with four domains was retained, but adapted or extended in detail, for example the ParSM presented in Gembarski [24] was integrated. In general, the entire system was built in an iterative process, as also provided for in VDI 2221 or MOKA [5, 29]. The four domains are elaborated as follows:

3.1.1. Customer Domain

The customer domain contains the customer requirements in informal form. An example for the customer needs for a skip loader are presented in Table *1*. This is the minimum set of customer needs necessary to configure a skip loader, although additional needs are possible. These needs are formulated informally and do not yet contain any concrete requirements. For example, the need "compatible with vehicle chassis XYZ" requires further information about vehicle XYZ, such as the options for designing and connecting the subframe.

Table 1. *Exemplary customer needs for a skip loader*

No.	Customer needs
1	Compatible with vehicle chassis XYZ
2	Payload of 7to
3	Possibility of container emptying
...	...

3.1.2. Functional Domain

In the functional domain, the customer needs of the preceding domain are converted into functional requirements. For this purpose, they are converted into a formal form by further deatilizing them and enriching them with knowledge if necessary. This formal form enables processing by the KBS. For the correct fulfillment of the customer needs, a larger number of functional reqirements usually follow from these. The functional reqirements are created in tabular form similar to a classic requirements list. In some cases, requirements are linked to values.

Table 2. *List of functional requirements*

No.	Functional requirement	Value
1.1	Frame length (mm)	4200 mm
1.2	Frame width	852 mm
1.3	Vehicle weight	3842 kg
...	...	
2.1	Hydraulics designd for load	7000 kg
2.2	Ensure stability during operation	
...	...	

3.1.3. Physical Domain

The physical domain contains all parameters of the product These are calculated based on the requirements of the previous domain. If further information is required for the calculation, this can be stored in tabular form, for example, so that the knowledge can be accessed. According to the basic architecture of KBS by Milton [9] this is called knowledge base while the calculations take place in the inference engine. Table *3* shows an example of a selection of parameters required to describe the main longitudinal beam of the skip loader's superstructure. Depending on the carrier vehicle, the size of the associated standard, which captured as a table in the knowledge base of he KBS, is selected. The form of the table shown here, with the columns name, value, unit and comment, corresponds to the layout of the parameter table in the CAD program used, Autodesk Inventor, so that the table can be adopted without restructuring. In total, more than 200 design parameters are needed to describe more than 90 single parts of the model of the skip loader superstructure. The naming is based on the corresponding subassembly and the corresponding component. in the example from table 3, »sf:ml:01« designates the first parameter of the main longitudinal beam of the subframe.

Table 3. *Selection of design parameters*

Name	Value	Unit	Comment
sf:ml:01	4200	mm	main longitud. beam: lenght
sf:ml:02	400	mm	main longitud. beam: height
sf:ml:03	110	mm	main longitud. beam: width
sf:ml:04	14	mm	main longitud. beam: wall thickness
...	...	...	...

3.1.4. Process Domain

The configurator presented here was designed only for research in the context of methodical product development in combination with design automation in the context of knowledge-based systems. Accordingly, process parameters for subsequent manufacturing were not developed, as actually intended in the axiomatic design. However, exemplary manufacturing constraints were established, which are intergated into the parameter space matrix presented in the following. These can be based on the existing machining tools. Examples for the skip loader examined here, which mainly consists of structural hollow sections, are the maximum cross section to be cut, the maximum manageable length of individual components or the maximum weight that can be lifted by an overhead crane. If these restrictions are stored centrally, they can be retrieved and embedded for different projects.

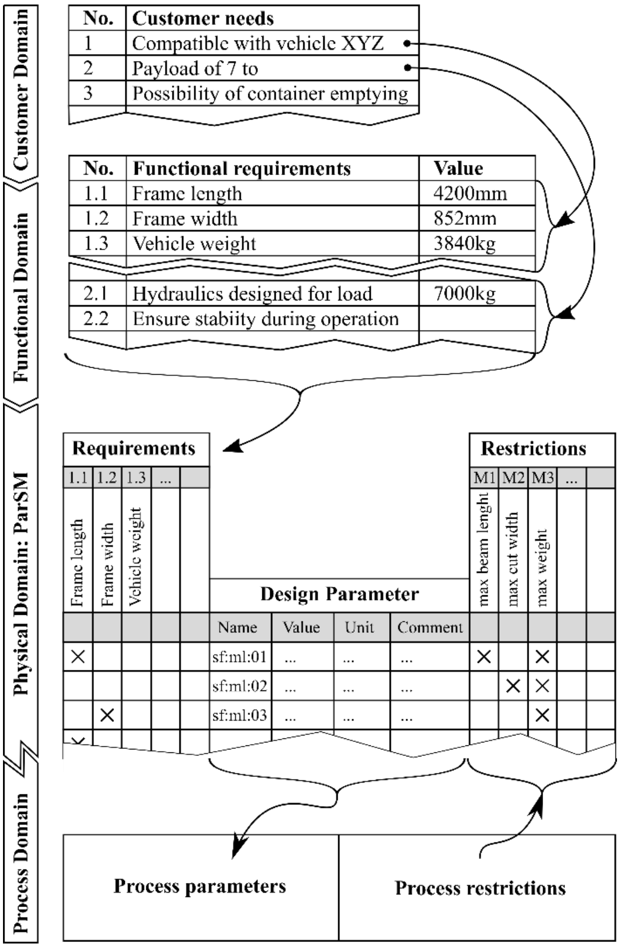

Fig. 6. *Axiomatic design adapted to the ParSM*

3.1.5. Parameter Space Matrix

For the configurator presented here, the ParSM is transferred to the axiomatic design approach. For this purpose, the ParSM combines the functional domain, physical domain and the process domain, as shown in the complete procedure model in Fig. *6*. However, the ParSM is not set up component-specifically, as shown by Gembarski [24], but globally for the entire assembly. This simplifies the process of parameter forwarding, since only the range of a table sheet is transferred once to the parameter list of the skeleton model in the CAD-system and from there is forwarded to the individual parts by derivation. Mainly, however, duplications are avoided, which arise when individual parameters are used for several components. The maintenance effort for changes is thus significantly lower than when using several ParSM, although the clarity also decreases when all parameters are linked to all requirements and restrictions in one table. However, a sensible grouping and naming of parameters, requirements and restrictions can help to counteract this. Using a combination of letters and numbers, parameters can be intuitively assigned to an individual part or subassembly.

Simple calculations can be performed directly within the cells of the ParSM. However, a large part is outsourced to another spreadsheet, as there is not enough space for some of the necessary intermediate steps in the ParSM or the clarity is lost. In addition, for some calculations the knowledge from further spreadsheets must be consulted. On these the dimensions and gradations of different standardized components are deposited. Furthermore, a vehicle database has been created, which contains the necessary characteristic values to develop the skip loader superstructure for a specific vehicle.

3.2. Axioms

In addition to the division into domains, the consideration of the two axioms is also necessary for the implementation of the axiomatic design approach. To fulfill the independence axiom, it is desirable to achieve an independence between functional requirements and design parameters, and thus to achieve an uncoupled design [7]. This is characterized by the fact that the design matrix A of the following equation is a diagonal matrix:

$$\{\boldsymbol{FR}\} = [\boldsymbol{A}]\{\boldsymbol{DP}\} \tag{1}$$

$$\begin{Bmatrix} FR_1 \\ FR_2 \\ FR_3 \end{Bmatrix} = \begin{bmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix} \begin{Bmatrix} DP_1 \\ DP_2 \\ DP_3 \end{Bmatrix} \tag{2}$$

If it is not a diagonal matrix, it is called a coupled system. According to Suh [7], a coupled system can be decoupled by converting the design matrix into a triangular matrix (cf. equation (3) and changing the order of the design parameters to allow row-by-row computation.

$$\begin{Bmatrix} FR_1 \\ FR_2 \\ FR_3 \end{Bmatrix} = \begin{bmatrix} A_{11} & 0 & 0 \\ A_{21} & A_{22} & 0 \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{Bmatrix} DP_1 \\ DP_2 \\ DP_3 \end{Bmatrix} \tag{3}$$

This is also referred to as quasi-coupled design. However, these assumptions have in common that the number of design parameters must be equal to the number of functional reqirements. If the number of design parameters is higher, the design is redundant.

However, as mentioned above, the CAD model of the skip loader superstructure contains a total of more than 200 design parameters due to the detailing chosen, which is significantly greater than the number of fuctional requirements. The dependencies are mostly redundant, which is also shown by the fact that the number of functional reqirements is significantly lower. This redundancy is deliberately chosen at this point, since otherwise the independence axiom can be fulfilled by reducing the number of design parameters, but the clarity of the listing of all parameters of the CAD model in the ParSM is lost. In addition, in the selected approach the fulfillment of the functional reqirements is not achieved by corresponding computations within the design matrix A, but by the already before mentioned outsourcing of complex computations. Nevertheless, by linking design parameters and functional requirements on the left side of the ParSM, the respective dependencies can be recognized. A dependency is marked by a cross in the corresponding row or column, as shown in Fig. *6*.

The fulfillment of the information axiom is much more complex. The goal is to select from different solutions the one that contains the least amount of information. In addition to the challenge of evaluating and measuring the information, it is necessary to develop more than one solution in order to be able to compare them. However, due to the high level of detail and complexity of the entire KBE system mentioned above, this is associated with a great deal of effort. In addition, a large number of iteration steps have to be run through for each solution. For this reason, the application of the information axiom is not feasible for the KBE example presented here.

3.3. Lessons learned: Limitations due to axiomatic design

During the development of the configurator as a KBE system using the axiomatic design methodology, some difficulties were encountered. The main problem is the complexity of the use case implemented here and its level of detail. In particular, the creation of a design matrix for the high number of functional requirements and design parameters is a challenge, so that this was not done in the way foreseen in axiomatic design. Instead, the ParSM approach was used and implemented in the domain structure. Unlike Gembarski [24], however, the ParSM was set up globally for the entire assembly and not on a component-specific basis. Thus possible duplications of parameters, which are needed in two components, are avoided. This reduces above all the error susceptibility in the case of maintenance at the system.

Furthermore, the process domain is devided into two parts. on the one hand, the domain contains the process parameters resulting from the design parameters. on the other hand, the process domain contains restrictions that limit the design parameters due to limitations in the existing manufacturing equipment. As a result, the solution space can be enlarged and more variants can be offered when the machine park is expanded and the process domain is updated.

All in all, this creates a clear tool for handling the parameters, but also the development of the requirements from the original customer needs is mapped in this way. With the increased complexity of a KBE system compared to a standard product with few variants, it is much easier to implement adjustments due to changed customer requirements or similar, since the dependencies and the information propagation within the system are clear.

4. DISCUSSION AND CONCLUSION

In this article, the structured handling of parameters within KBE system was discussed starting from customer requirements up to the manufacturing process. To this end, existing methodologies for the development of KBE system were first presented, compared and evaluated with regard to their suitability for a product configurator. Subsequently, the axiomatix design methodology was presented together with the associated axioms and domains, and the parameter space matrix was introduced. On this basis, a procedure model is then developed, which integrates the ParSM into the domain structure of the axiomatic design. This was done by means of the development of a configurator for the design of skip loader superstructure for trucks. The limits of the axiomatic design for a practicable application were shown by the structure of this KBS. In particular, the use of the design matrix in strict compliance with the independence axiom is difficult at the level of detail that was selected here. Furthermore, due to necessary calculation tasks, such as the development of the kinematics of the swivel arms, a calculation of the parameters by means of matrix multiplication was not feasible. Furthermore, the information axiom was not applied, since a comparison and thus the development of several independent KBE system is necessary to determine the solution with the lowest information content. This conflicts with the usual iterative problem-solving strategy, as described for example in VDI 2221 or MOKA [5, 29].

Based on this experience, a process model was developed that combines the division of the development process into domains according to axiomatic design with the linking of parameters with requirements and process restrictions of the ParSM. This enables clear handling of a large number of parameters. The dependencies are also clear across the boundaries of the domains, so that adjustments to customer requirements can be made without great effort. This makes the process model developed particularly suitable for KBS and for the further use of stored knowledge, as this is only possible if the relevant systems can be operated and maintained independently of people. Furthermore, the ParSM serves as a central interface between all calculation steps, which makes it possible to subdivide the overall problem into subproblems. This results in individual tasks that are easier to handle. In addition, solutions of subtasks can also be used independently of the project for the development of other KBS.

In addition to the implementation of the product configurator described here with the developed process model consisting of axiomatic design and ParSM, the MOKA methodology was applied in a parallel project. The six phases defined by Stokes [5] were run through and a configurator with the same range of functions was developed. It was found that the two methodologies support the developer to different degrees in different phases of the product development process. MOKA mainly provides guidelines for the knowledge acquisition and implementation steps and takes a holistic view of the process of developing a KBS.

The methodology developed in this article, on the other hand, focuses mainly on the handling of parameters, which is essential in the case of KBE system development. This is particularly important for tasks with a high level of detail and to ensure maintainability. Furthermore, although the methodology does not provide concrete specifications for knowledge acquisition, it goes beyond the scope of MOKA with regard to the definition of process restrictions and parameters. It should also be added that the »axiomatic ParSM« does not compete with methodologies such as MOKA, but can also be integrated into them. For example, the documentation provided in MOKA by means of ICARE forms can be dispensed with and the tabular form presented here can be chosen instead. This is particularly useful for projects with a high level of parametric design. In contrast to knowledge storage by ICARE, ParSM is very application-oriented and can usually be implemented with existing tools, i.e. Excel and a CAD program.

In the further course of the research, the combination of the axiomatic design in conjunction with ParAM methodology presented here can be tested and evaluated on the basis of further examples. The direct derivation of the process parameters from the KBE system in particular offers a lot of potential with regard to the core idea of the KBS, namely the avoidance of routine activities.

6. REFERENCES

[1] K. Kristjansdottir, S. Shafiee, and L. Hvam, "How to identify possible applications of product configuration systems in Engineer-to-Order companies," *International Journal of Industrial Engineering and Management, vol.8*, no. 3, pp. 157–165, 2017.

[2] C. Lutz, *Rechnergestütztes Konfigurieren und Auslegen individualisierter Produkte: Rahmenwerk für die Konzeption und Einführung wissensbasierter Assistenzsysteme in die Konstruktion*. Zugl.: Wien, Techn. Univ., Diss., 2011, 1st ed. München: Verl. Dr. Hut, 2012.

[3] W. J. Verhagen, P. Bermell-Garcia, R. E. van Dijk, and R. Curran, "A critical review of Knowledge-Based Engineering: An identification of research challenges," *Advanced Engineering Informatics*, vol. 26, no. 1, pp. 5–15, 2012.

[4] G. Pahl, W. Beitz, J. Feldhusen, and K.-H. Grote, *Engineering Design*. London: Springer London, 2007.

[5] M. Stokes, Ed., *Managing engineering knowledge: MOKA: methodology for knowledge based engineering applications*. London, Bury St. Edmunds, 2001.

[6] G. Schreiber *et al.*, *Knowledge Engineering and Management: The CommonKADS Methodology*: The MIT Press, 1999.

[7] N. P. Suh, *The principles of design*. New York: Oxford Univ. Press, 1990.

[8] K. Kurbel, *Entwicklung und Einsatz von Expertensystemen: Eine anwendungsorientierte Einführung in wissensbasierte Systeme,* 2nd ed. Berlin, Heidelberg: Springer, 1992.

[9] N. R. Milton, *Knowledge technologies*. Monza: Polimetrica, 2008.

[10] P. C. Gembarski, "Three Ways of Integrating Computer-aided Design and Knowledge-based Engineering," *Proc. Des. Soc.: Des. Conf.*, vol. 1, pp. 1255–1264, 2020.

[11] A. A. Hopgood, *Intelligent systems for engineers and scientists: A practical guide to artificial intelligence*. Boca Raton, FL: CRC Press, 2022.

[12] W. Skarka, "Application of MOKA methodology in generative model creation using CATIA," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 5, pp. 677–690, 2007.

[13] T. Brockmöller, R. Siqueira, P. C. Gembarski, I. Mozgova, and R. Lachmayer, "Computer-Aided Engineering Environment for Designing Tailored Forming Components," *Metals*, vol. 10, no. 12, p. 1589, 2020.

[14] O. Cairó and J. C. Alvarez, "The KAMET II Approach for Knowledge-Based System Construction," in *Lecture Notes in Computer Science, Knowledge-Based Intelligent Information and Engineering Systems*, D. Hutchison et al., Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 1227–1234.

[15] J. Angele, D. Fensel, D. Landes, and R. Studer, "Developing Knowledge-Based Systems with MIKE," *Automated Software Engineering*, vol. 5, no. 4, pp. 389–418, 1998.

[16] R. Curran, W. J. Verhagen, M. J. van Tooren, and T. H. van der Laan, "A multidisciplinary implementation methodology for knowledge based engineering: KNOMAD," *Expert Systems with Applications*, vol. 37, no. 11, pp. 7336–7350, 2010.

[17] A. Tedeschi and D. Lombardi, "The Algorithms-Aided Design (AAD)," in *Informed Architecture*, M. Hemmerling and L. Cocchiarella, Eds., Cham: Springer International Publishing, 2018, pp. 33–38.

[18] J. Cagan, M. I. Campbell, S. Finger, and T. Tomiyama, "A Framework for Computational Design Synthesis: Model and Applications," *Journal of Computing and Information Science in Engineering*, vol. 5, no. 3, pp. 171–181, 2005.

[19] R. F. Riesenfeld, R. Haimes, and E. Cohen, "Initiating a CAD renaissance: Multidisciplinary analysis driven design," *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 1054–1072, 2015.

[20] M. Biedermann and M. Meboldt, "Computational design synthesis of additive manufactured multi-flow nozzles," *Additive Manufacturing*, vol. 35, p. 101231, 2020.

[21] P. Müller, P. C. Gembarski, and R. Lachmayer, "Parametric Topology Synthesis of a Short-Shaft Hip Endoprosthesis Based on Patient-Specific Osteology," in *Lecture Notes in Mechanical Engineering, Towards Sustainable Customization: Bridging Smart Products and Manufacturing Systems*, A.-L. Andersen et al., Eds., Cham: Springer International Publishing, 2022, pp. 669–676.

[22] O. Kulak, S. Cebi, and C. Kahraman, "Applications of axiomatic design principles: A literature review," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6705–6717, 2010.

[23] N. P. Suh, *Complexity: Theory and applications*. Oxford: Oxford University Press, 2005.

[24] 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," *IJIEM*, vol. 9, no. 4, pp. 177–186, 2018.

[25] P. C. Gembarski, H. Li, and R. Lachmayer, "KBE-Modeling Techniques in Standard CAD-Systems: Case Study—Autodesk Inventor Professional," in *Springer Proceedings in Business and Economics, Managing Complexity*, J. Bellemare, S. Carrier, K. Nielsen, and F. T. Piller, Eds., Cham: Springer International Publishing, 2016, pp. 215–233.

[26] *Multi-bucket system vehicles, swivelling devices: Part 1: For standard containers with a nominal volume of 5,5 m³ to 20 m³*, DIN 30723-1:2016-02, Beuth, Berlin, 2016.

[27] K. Roth, *Konstruktionslehre*, 3rd ed. Berlin: Springer, 2000.

[28] D. A. Gebala and N. P. Suh, “An application of axiomatic design,” *Research in Engineering Design*, vol. 3, no. 3, pp. 149–162, 1992.

[29] *Design of technical products and systems: Model of product design*, VDI 2221-1:2019-11, VDI, Berlin, 2019.

CORRESPONDENCE

Marcus Oel, M.Sc
Institute of Product Development,
Leibniz University Hannover,
An der Universität 1
30823 Garbsen, Germany
oel@ipeg.uni-hannover.de

Dr.-Ing. Paul Christoph Gembarski
Institute of Product Development,
Leibniz University Hannover,
An der Universität 1
30823 Garbsen, Germany
gembarski@ipeg.uni-hannover.de

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