INTEGRATING SALES AND DESIGN: APPLYING CAD CONFIGURATORS IN THE PRODUCT DEVELOPMENT PROCESS

L. P. Poot ✉, C. Wehlin, M. Tarkian and J. Ölvander

Linköping University, Sweden

✉ leon.poot@liu.se

Abstract

With industries striving towards increased customisation of complex products through engineer-to-order, methods are continuously sought to rationalise the product development process. To this end, a framework is proposed using CAD configurators, utilising design automation and knowledge-based engineering to integrate sales and design processes in product development. The application of this framework to the design of spiral staircases is described and analysed, with results showing decreased lead-times and a decreased risk for design errors.

Keywords: computer-aided design (CAD), design automation, knowledge-based engineering (KBE), product configuration, mass customisation

1. Introduction

One of current trends in the industry is the strive towards increased customisation of products. This leads companies to continuously seek methods of rationalising the product development process. In a mass customisation or engineer-to-order (ETO) setting, engineering activities demand a considerable amount of development resources for each customer. A large part of these activities involves repetitive design tasks, which are time-costly and prone to errors. Rationalisation of such design tasks can be achieved through design automation (DA) by utilising product knowledge through knowledge-based engineering (KBE).

Arguably, all unnecessary costs can be attributed to engineering errors occurring during the development process. It is therefore crucial to catch and mitigate these errors as early as possible in the process of sales, development and manufacturing. This shows the need for flexibility and correct communication ensured through the use of intuitive tools and methods. Oftentimes, crucial customer requirements are lost in translation due to departments essentially speaking "different languages". This can result in costly mistakes being caught far later than needed, and in a worst-case scenario render a product unsaleable. Streamlining the communication between sales and design departments by effectively bridging engineering language gaps is an essential step towards integration throughout the product development process, and thereby rationalising the entire process required to provide complex, mass-customised and customer-unique products.

The product development process consists of several design phases. Regardless of the methodology of choice these boil down to task analysis (customer) followed by conceptual design, embodiment and detailed design (in this paper combined under detailed design), and an implementation phase, typically production (Howard et al., 2008). In a mass customisation setting, where these phases can be repeated throughout a products lifecycle, development of a product's variants can be considered as standalone

development processes. A large part of the activities within these phases involve computer-aided design. In its conceptual phase, configurators are a set of tools that can aid the design of specific variants (Trentin et al., 2012).

A prime example of an industry characterised by a high level of customisation is the staircase industry. Tailor-designed for private, public and company buildings, requirements vary wildly between environments ranging from the primarily aesthetic to emergency exits. Coupled with the relatively complex design process of spiral staircase variants, this poses a challenge for companies competing with low-cost manufacturers. Piller and Tseng (2009) describe the case of a manufacturer of handcrafted wooden staircases strengthening their position in the industry by standardising components yet retaining direct contact with the customer early in the design process. This is accomplished through use of (online) configurators, building on the standardisation of their product.

With the aim to rationalise the product development process and support mass customisation, this article describes an interactive framework designed to include the customer in the early design stages through configurators, utilising DA through KBE as presented by (Amadori et al., 2012). The framework is implemented as a proof-of-concept in the industrial case of mass-customised staircases in cooperation with a leading Swedish stair manufacturer.

2. Background

2.1. Mass customisation

Mass customisation has traditionally been enabled through product platforms and a configure-to-order (CTO) approach, where products are built using pre-developed building blocks, which restricts the way in which customer requirements can be met. An engineer-to-order (ETO) approach includes customers earlier in the engineering process, during design of the product, and thus allows for a higher level of customisation according to Levandowski et al. (2015). Whereas a traditional ETO design process relies entirely on manual design tasks due to the uniqueness of products, ETO can be applied to parts of a product of which the design process can be applied in a mass customisation setting (Haug et al., 2009).

In a manufacturing value chain, the customer order decoupling point (CODP) is defined as the point where a product is linked to a specific customer order. This results into one of the following processes (in decreasing order of customisation level): engineer-to-order (ETO) which includes the customer at the earliest possible point, manufacture-to-order (MTO), assembly-to-order (synonymous with CTO), and manufacture-to-stock (MTS) which relies entirely on retail activity. Manufacturers can affect reliability and lead times by prefabricating components, moving the CODP from MTO to CTO, requiring certain standardisation of the products' components. This however causes greater uncertainty. To accommodate a higher level of customisation and required design activities the CODP should be moved upstream. According to Wikner and Rudberg (2005), this is not a realistic view and it is in fact a two-dimensional problem: whereas all four terms are originally linked to production only, ETO stands out as a special case of MTO with the added dimension of engineering design, which allows the engineering activities for customisation to occur parallel with production.

2.2. Configurators

Implementation of mass customisation is often approached through product configurators, aiming at translation of customer requirements to the product information while adhering to pre-set rules for product variants, (Salvador and Forza, 2004). The use of product configurators dates back to the computer system configuration program R1 (McDermott, 1982), evolving into a field of considerable interest in both industry and academia (Zhang, 2014). Not only must a successful product configurator handle conceptual modelling of configurations, but also recommendation and solving of such configurations and crucially allow for reconfiguration. Product configurators can never directly support complete customisation due to their being based on predefined functions and components. As such, it is impossible to prepare for all unforeseen customer requirements (Zhang, 2014). This is comparable to the balance between repetitive, automatable tasks and creative, innovative and value-adding tasks. However, a flexible product configurator allows both configuration of predefined modules (i.e. a combinatorial

problem with a fixed number of possible combinations), as well as modification of module parameters yielding a large solution space. A product configurator must also allow for introduction of new components with relative ease for continued success. The size of the solution space must be kept in balance, and continuous oversight is required so that the implementation remains profitable (Gorski et al., 2016). Maintenance of configurations, knowledgebases and the system itself is essential. Therefore, flexibility is a key factor (Haag, 1998; Trentin et al., 2012).

A product configurator may focus purely on engineering design but can have great advantages when applied in a sales context, where multiple sub-systems and loops can be involved in the configuration process for complex products. This will be a configurator on a higher abstraction level as a sales configurator typically involves external problem solvers, i.e. salespersons or consumers who do not necessarily have knowledge about the technical aspects of a product but understand its function (Haag, 1998). Product/sales configurators form the base for what is known in the industry as Configure Price Quote (CPQ) platforms, a term which is infrequently encountered in academia however (Sorri et al., 2017). The use of product configurators has been shown to provide a significant decrease in required working hours per product of up to 75%, as well as considerably shortened lead times (Kristjansdottir et al., 2018).

2.3. Design automation

CAD has, since Ivan Sutherland introduced SKETCHPAD (Sutherland, 1963), had continuous impact on product development, to the point where 3D modelling is now widely accepted throughout the industry. Whereas it has helped the creation, analysis and production of increasingly sophisticated and complex products, approximately 80% of CAD design tasks are still repetitive and routine-like (Stokes, 2001). Not only is this time-costly, it also allows errors to creep into the designs due to the essentially tedious nature of these tasks (Tarkian, 2012).

Through DA these repetitions can be brought to a minimum, thus rationalising the design process. A means of achieving this is KBE, creating systems allowing utilisation of existing knowledge to create new products or variants of existing products. The goal of KBE is to automate the aforementioned repetitive tasks, and subsequently support reuse of knowledge and multidisciplinary optimisation through all stages of the design process (Rocca, 2012).

A practical application of DA in CAD is the use of High-Level CAD templates (HLCt), (Amadori et al., 2012). Both geometrical and non-geometrical knowledge are stored in the templates themselves as opposed to the knowledgebase central to the system. This ensures that knowledge is kept close to its source, the components. The geometry and its governing set of parameters form the base, with contextual knowledge being integrated into the template. This includes specific links between component templates as well as relevant conditions and information specifically required by the configurator for operations such as naming and replacement of components. These templates are used in conjunction with a CAD configurator as building blocks, rendering the user of such system able to build products in a 3D CAD environment. The HLCt extend past low-level automation through parametric modelling and allow both morphological and topological transformations, i.e. changes in both in the shape of a component and its contextual relations (Amadori et al., 2012).

Such a CAD configurator is a DA framework realized through a knowledge-based system (KBS), where an inference engine executes knowledge stored in a knowledgebase which, through a HLCt database, enables generation of complex CAD models controlled by a user through an interface, based on contextual knowledge collected from the templates before execution (Tarkian, 2012). This is comparable to object-oriented programming, albeit in a geometric context. DA in general and configurators in particular allow for rapid, iterative generation of concepts. This enables simulation and optimisation applications, facilitating acquirement of more knowledge earlier in the development process and thus allowing greater design freedom (Verhagen et al., 2012).

While DA is effective in minimising repetitivity, it does not solve the issue of finding the best solution to a design problem. To that end, the aim of design optimisation is to find the optimal design of a product by identifying design characteristics to satisfy design requirements. An engineering problem will often encompass multiple conflicting objectives, resulting in an inescapable trade-off. The best trade-off solutions, known as Pareto-optimal designs, will be equally good yet satisfy the objectives to varying degrees (Chang, 2015). A popular method for optimisation of complex design problems is the

application of genetic algorithms, a robust optimisation technique which mimics natural selection. Through multiple generations, a population of individual designs is evaluated, the best of which are chosen to form the base of each next generation, resulting in a Pareto-optimal set (Andersson, 2000).

3. Proposed framework

To maximise involvement of customers and minimise repetitive tasks in mass customisation of complex products, a framework is proposed consisting of tools utilising KBE aimed at DA. A customer is directly involved in the conceptual design of variants, while the effort required is limited to input of the minimum required customer requirements.

3.1. Framework layout

The proposed framework utilises configuration in different levels of abstraction, depending on the maturity of the design. At the heart of this expert system lies a dual configuration system, consisting of a sales configurator in conjunction with a design generator tool and a technical product (CAD) configurator. An initial iteration cycle for conceptual design involves a sales configurator interface, which together with the design generator module guides its user towards a final concept based on a set of requirements. The larger design cycle includes the product or CAD configurator, from which quotes can be produced for order acquisition. Aside from controlling the internal configuration process, the sales configurator furthermore functions as a means to convey and retrieve information to and from the user through its interface.

As seen in Figure 1, the process spans multiple design phases. The framework focuses on development of variants of an existing product. Spanning the product development process from conceptual to detail design and production preparation, it aims to provide a means to achieve an automated and flexible mass customisation process. Whereas today the customer provides requirements and is thus in a way involved from the very beginning, the process of designing a custom staircase removes the customer from the equation until a quote for the final design can be presented, risking prolonged iterations which may take days or weeks. The proposed framework aims to shorten this timeframe significantly, allowing a preliminary quote to be delivered to the customer virtually instantaneously.
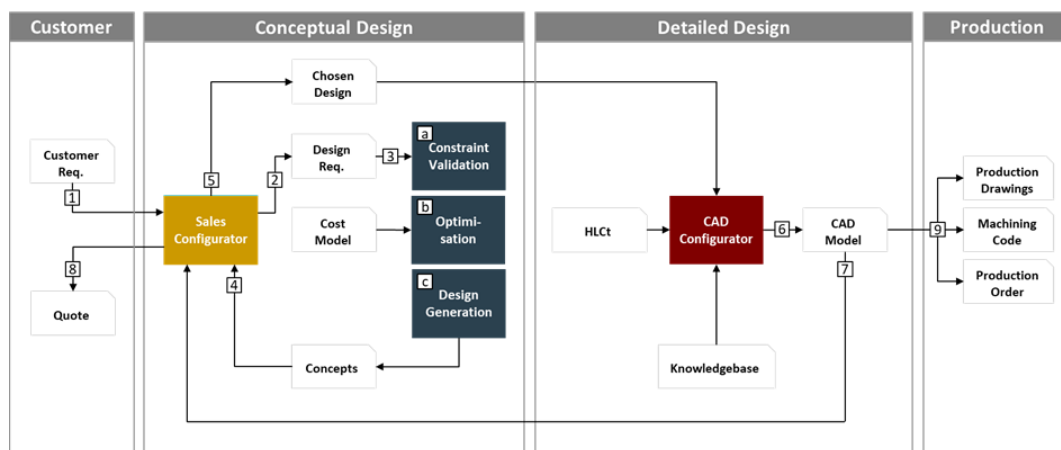

Figure 1. Proposed framework for mass customisation of complex products

KBE is applied in both the conceptual design and detail design phase shown in Figure 1. A knowledgebase incorporated in the design generator allows automated generation and cost-estimation of designs based on product-specific rules and data. While in the detail design phase the knowledgebase together with the HLCt database allows the inference engine behind the CAD configurator to automate design tasks and thus generate accurate CAD models of the configuration. The sales and CAD configurator thus act in a similar manner. The primary difference being that no CAD geometry, nor knowledge related to the generation of such geometry comes into play in the conceptual stage of the framework, which eliminates the computationally demanding process of solving the configuration in detail and building the geometry entirely. This results in a shorter lead time per design generation, albeit at the cost of fidelity.

The shortened lead time per design opens up for inclusion of a DA framework generating a large number of designs. Combined with an optimisation framework favouring certain designs or design traits based on user input, this becomes a powerful tool for rapid design iteration and simultaneous evaluation. Hence, in the same amount of time, a larger set of designs could be presented to the customer. Thus, the sales configurator is a means of increasing the amount of acquired knowledge at an early stage, while retaining a significant amount of design freedom.

3.2. Implementation

In order to test the applicability of the proposed framework, it has been applied in a case study concerning mass customisation of staircases. The case of spiral staircases addressed in this study shows a high level of complexity in the design process of an everyday product. Industrially produced as opposed to handcrafted, the design and production of the staircase are required to fulfil certain standards while minimising costs. For example, safety requirements dictate a minimum head clearance, minimum railing height, and minimum stair width or radius, among others. These requirements vary between countries due to national legislature, and therefore between orders for an internationally active company.

Besides the strict safety requirements, company guidelines may impose further restrictions. Absolute safety limits act as design constraints, on which rules in the knowledge-based system are built, while design parameters may rather act as target values in finding an optimal stair design from a usability perspective. Examples of parameters affecting the usability are, as shown in Figure 2: stair radius (r), step height(h), step angle (α), landing angle (β) and head clearance (H). The framework allows the user to specify constraint values as settings for a design feasibility check, while target values for design requirements can be entered directly in the sales configurator's interface.
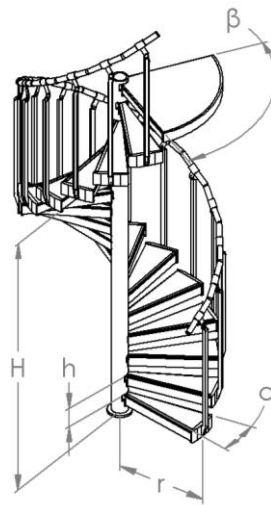

Figure 2. Spiral staircase parameters affecting usability

As described in section 3.1, the framework uses an optimisation module for generation and automatic evaluation of concepts before relaying these to the user through the sales configurator. This results in the following process, where the numbers refer to the numbered arrows in Figure 1:

1. The sales configurator takes customer requirements, provided by either a sales representative or directly by a potential customer. An interface guides the user through the minimal required input for a design. The user is prompted to provide the requirements by choosing options or entering specific values or ranges for the desired stair characteristics, as well as constraints such as those dictated by applicable safety requirements.

2. The customer requirements are translated to design requirements and written to an external file in a predefined XML-format. The design requirements include design parameter ranges corresponding to the dimensions shown in Figure 2 and values describing design choices such as tread material, preferred type of railing and surface treatment.

3. Several designs are generated, resulting in approximations of required production time and cost estimates as well as design configuration specifications for later detailed modelling.

   a. The design requirements are in an initial feasibility check analysed to fulfil constraints by means of a mathematical model approximating the resulting geometry.

   b. In the optimisation module, a number of designs are produced within the ranges allowed by the feasibility check and design requirements. By means of a genetic algorithm these are optimised by weighting cost versus a usability factor. The cost is based on approximative calculations of the amount of material required and appurtenant manufacturing costs combined with a pricelist of standard components, stored in a built-in knowledgebase. The usability factor is calculated based on the geometrical properties of the design weighted against pre-defined goal values.

   c. Amongst the best-fitting designs from the optimization, a limited set is selected, evenly distributed along the resulting Pareto front in order to provide a complete overview of the trade-off. For each, a bill of material (BOM) and configuration specification is generated and written in an XML-format tailored to fit the CAD configurator interface.

4. The result is fed back to the interface for the sales representative and/or customer to examine. Values describing the cost, entry and exit angles for each flight, vertical head clearance and other decision factors for each design are fed back directly to the sales configurator. For each design, an XML file is loaded, and a lightweight graphical representation is generated within the user interface.

Steps 2-4 are carried out relatively quickly with the process appearing near-instant to the user, enabling quick iterations on the fly until the conceptual design is deemed satisfactory by the customer. At this point, the conceptual cycle is completed. All concept fed back to and presented in the sales configurator are taken from the Pareto front and therefore optimal, leaving it up to the user to decide on the importance of each objective prioritising usability over cost or vice versa. The cycle renders new sets of concepts within seconds, allowing for continuous customer contact or involvement throughout iterations. Once a design is chosen for continuation, the framework continues to generate it in detail:

5. The configuration specification for the chosen design is transferred to the CAD configurator, which, through its inference engine and knowledgebase, translates these to the required design parameters.

6. An inference engine in conjunction with a model database and a knowledgebase translates the required design parameters for a detailed design and builds a CAD model. Bespoke CAD models are instantiated for each part described in the XML from the HLCt and assembled. The design parameters of the instantiated components are automatically adjusted according to the specifications. For orders requiring components to locally deviate from the standard templates, the design parameters can be modified manually at this stage.

7. Based on the detailed CAD model and knowledge contained within the templates and knowledgebase, precise cost estimations and visualisation material are generated, fed back to the sales configurator for examination.

From the detail design, a more accurate model (or depictions thereof) and decision basis are taken back to the sales configurator, where the user can verify that the design fits the customer requirements. If required, another of the Pareto designs can be selected for detail design, or the design requirements can be modified in the sales configurator for reiteration. Due to the level of detail, steps 5-7 require considerably longer time than the conceptual iteration. Once the detailed design is satisfactory to the customer, the design can be greenlit for production.

8. From the sales configurator, a preliminary quote is generated for the customer based on the required components and their individual costs.

9. From the CAD model, manufacturing and production data for the final design may be generated in the form of production orders, production drawings, assembly instructions and machining code.

At this point, the order can be finalised, and production begin.

3.3. Proof-of-concept and validation

3.3.1. Conceptual design

A demonstrator acting as a proof-of-concept of the framework has been developed together with the aforementioned stair manufacturer. For testing purposes, the conceptual design cycle consists of a rudimentary sales configurator interface designed using Microsoft Excel allows the user to input the requirements. These are read by a Python script, after which the optimisation using an implementation of a NSGA-II algorithm with the open-source framework DEAP (Fortin et al., 2012) takes over. Five designs are chosen from the resulting Pareto front and written to the sales configurator. The values for each design variable are shown and visualised by means of pie charts depicting the starting and ending angles of each flight. A conceptual sales configurator has shown that a user can, without a deep understanding of stair design, be guided through the inputs to produce multiple conceptual designs, and near instantly have these designs generated and the results shown. The proof-of-concept demonstrator is visualized in Figure 3, showing the input form together with an example of the presented results.

Figure 3. Proof-of-concept demonstrator for sales configurator with input (left) and result output (right) fields

3.3.2. Detail design

As the conceptual design cycle is completed, detail design requires output data for the chosen concept to be adapted to the CAD configurator. For this purpose, the description of the stair design is written to a configuration specification in XML format for all CAD parameters of all parts to be modelled. Information concerning each part's type of component is included. As parts build on each other in context, information concerning their parent parts is given. Each part is assigned a unique ID, to which the parameters refer. The configuration specification is imported into the CAD configurator linked to the HLCt database. For this study, templates were modelled in SOLIDWORKS and used to generate staircase designs with the configuration software XperDi CAD Configurator. These consist of parametric geometry controlled using equations for morphological changes, built on reference geometry from parent templates and providing the same for child templates. Specific naming of parameters and reference geometry allows the CAD configurator to instantiate the models in context, while a template in itself has no direct contextual links to other CAD models. An example of this is shown in Figure 4.
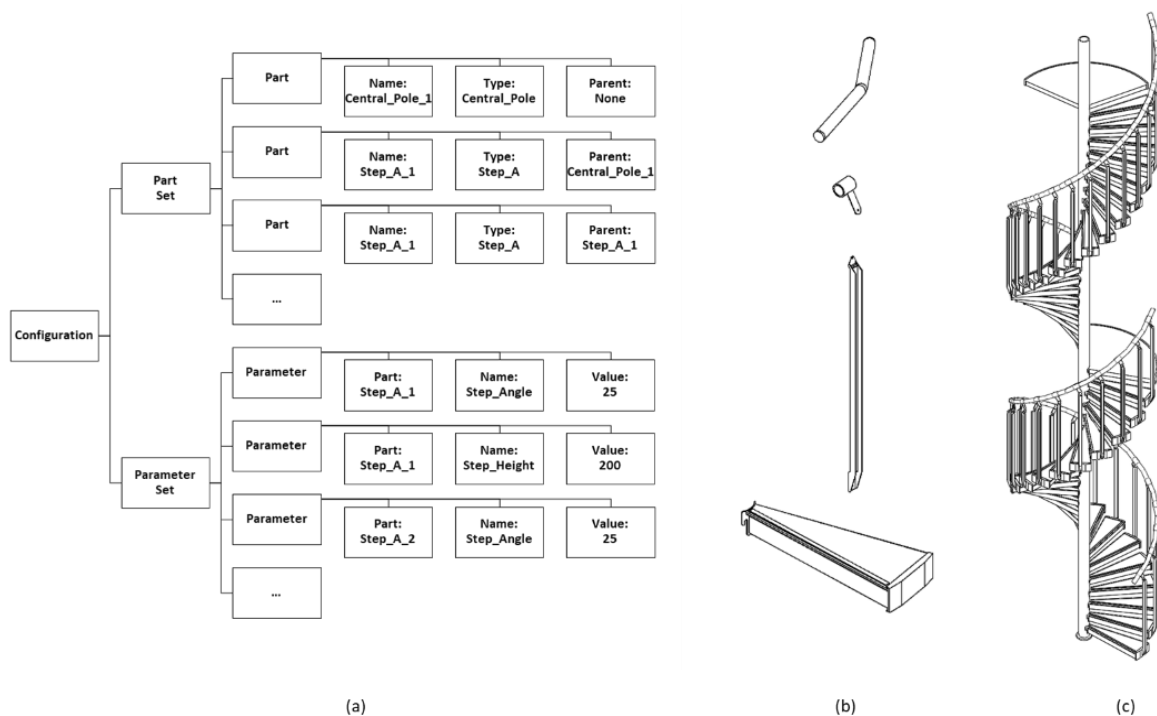
Figure 4. Steps of CAD model generation, with a) a simplified configuration specification of required components and parameter values, b) HLCt templates, and c) a generated CAD model

Templates are instantiated sequentially, with the central pole acting as a base for steps, steps supporting other steps or balusters, which in turn provide support for railing segments. A combination of parametrically modifiable and standard models is used as templates. The database consists of 36 templates. 18 of these are modified morphologically for each instantiation, whereas the remaining correspond to standard step components, mirroring a list of standard components included in the cost model for accurate cost estimation as opposed to dimension-based approximations off material and manufacturing costs. From the resulting model, production data can be generated using CAD/CAM or other external software.

The CAD configurator allows not only generation but also modification of the design through its interface. This may be achieved through morphological transformations, i.e. modifying dimensions of certain components to e.g. accommodate unique parts, or topological transformations such as the replacement of a component with another while retaining the contextual links with the original parent and child components.

3.3.3. Implementation validation

The demonstrator has been evaluated together with the stair manufacturer, comparing the results with stairs designed with the traditional manual design approach. It was concluded that the demonstrator was able to handle customer requirements for approximately thirty cases labelled as being complex by the company and generate designs very similar to those created by experienced engineers in a fraction of the time that was needed by the traditional design approach. Direct translation of design requirements and the contextual links included in the HLCt templates ensure that all components are sized and positioned correctly according to customer requirements. Hence the demonstrator shows great potential in enabling mass customisation with fast iterations and fewer errors. Results rendering a design undesirable were found to be fewer than expected compared to manually engineered solutions and were solved by reiterating with narrower requirements. With the framework, it now takes approximately 1-3 hours to configure and generate a spiral staircase composed of 2-3 flights, including iterations with the customer as well as automated engineering tasks. This means the process is sped up by an order of magnitude compared to using the traditional approach of iterating on customer requirements and concepts manually.

4. Discussion and conclusion

A framework is presented aiming to provide a means to rationalise the product development process in mass customisation, integrating all phases from customer contact to production preparation through design automation achieved using knowledge-based engineering systems. The conceptual framework speeds up the design process requiring less development resources, while decreasing the risk for design errors in a mass customisation setting. It can be applied to the design of variants of any product involving complex development processes requiring extensive, repetitive CAD modelling.

Product configurators will inevitably limit design freedom and can therefore never support complete customisation. The level of standardisation will thus define its CODP on the CTO-ETO spectrum. A custom product consisting entirely of predefined components can be seen as CTO. As components contain more and more morphological redesign the CODP of the product drifts towards ETO. Components of the spiral staircase design studied require both topological and morphological changes, putting the CODP of this particular product in between CTO and ETO. Often, variants of these components generated using the CAD configurator are entirely new in shape. Their function, however, remains the same. The aim of the framework with its sales/CAD configurator setup is to allow customers to configure their unique product with the help of product experts where needed. The need for CAD expertise is brought to a minimum, required only in such case where entirely unique components are required for ETO.

With ETO seen as a two-dimensional problem (Wikner and Rudberg, 2005), the framework takes care of the engineering dimension while maintaining flexibility in production. The parallel use of both standard and unique, customised components in the configurator allows for both assemble- and manufacture-to-order, or a combination thereof. With HLCt templates, numerous standardised variants can easily be included in the database as well as modifiable templates. By balancing between standard and customised components, a user continues to have control over their CODP and design freedom.

In the study on spiral staircase design, optimisation is based on a mathematical approximation of stair geometry and a list describing approximated production costs for available components. Designs and the resulting quotes must be kept relevant, and so the templates and models must be maintained. The use of enterprise resource planning software may be the key to this required maintenance, while ensuring direct access to accurate cost figures and stock balances for standard components. Generation of models may be sped up by storing and reusing generated concepts. This can also provide an additional aspect to optimisation, allowing prioritisation of pre-existent designs to eliminate the need of generation and verification of models and thereby cutting development costs.

An aspect for future consideration is the layout and functions of the sales configurator user interface. While the demonstrator presented here shows a basic, numerical representation of the result, concepts shown directly in the configurator as a lightweight 3D representation will enhance the comprehensibility of the designs' strengths and elevate the overall user experience. For this, game engines can be used due to their fast rendering capabilities. On the other end, the nature of HLCt templates with the possibility of including non-geometric knowledge opens up for automatic generation of production data. As all components produced by the configurators will be variants of a known design, their production preparation can be completely automated once the general production method for these has been defined. This is key in rationalising the entire process from first consumer contact to production and is of interest for further research in the implementation of the framework.

Acknowledgements

References

Amadori, K. et al. (2012), "Flexible and robust CAD models for design automation", *Advanced Engineering Informatics*, Vol. 26 No. 2, pp. 180-195. https://doi.org/10.1016/j.aei.2012.01.004.

Andersson, J. (2000), "A survey of multiobjective optimization in engineering design", *Optimization*, pp. 34. https://doi.org/10.1.1.8.5638.

Chang, K.-H. (2015), "Multiobjective Optimization and Advanced Topics", In: *Design Theory and Methods Using CAD/CAE*, pp. 325-406. https://doi.org/10.1016/b978-0-12-398512-5.00005-0.

Fortin, F.A. et al. (2012), "DEAP: Evolutionary algorithms made easy", *Journal of Machine Learning Research*, Vol. 13, pp. 2171-2175.

Gorski, F., Zawadzki, P. and Hamrol, A. (2016), "Knowledge based engineering as a condition of effective mass production of configurable products by design automation", *Journal of Machine Engineering*, Vol. 16 No. 4, pp. 5-30.

Haag, A. (1998), "Sales configuration in business processes", *IEEE Intelligent Systems and Their Applications*, Vol. 13 No. 4, pp. 78-85. https://doi.org/10.1109/5254.708436.

Haug, A., Ladeby, K. and Edwards, K. (2009), "From engineer-to-order to mass customization", *Management Research News*, Vol. 32 No. 7, pp. 633-644. https://doi.org/10.1108/01409170910965233.

Howard, T.J., Culley, S.J. and Dekoninck, E. (2008), "Describing the creative design process by the integration of engineering design and cognitive psychology literature", *Design Studies*, Vol. 29 No. 2, pp. 160-180. https://doi.org/10.1016/j.destud.2008.01.001.

Kristjansdottir, K. et al. (2018), "Return on investment from the use of product configuration systems – A case study", *Computers in Industry*, Vol. 100, pp. 57-69. https://doi.org/10.1016/j.compind.2018.04.003.

Levandowski, C.E., Jiao, J.R. and Johannesson, H. (2015), "A two-stage model of adaptable product platform for engineering-to-order configuration design", *Journal of Engineering Design*. https://doi.org/10.1080/09544828.2015.1021305.

McDermott, J. (1982), "R1: A rule-based configurer of computer systems", *Artificial Intelligence*, Vol. 19 No. 1, pp. 39-88. https://doi.org/10.1016/0004-3702(82)90021-2.

Piller, F.T. and Tseng, M.M. (2009), *Handbook of research in mass customization and personalization, Handbook of Research in Mass Customization and Personalization*, World Scientific Publishing Company. https://doi.org/10.1142/P7378.

Rocca, G.L. (2012), "Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design", *Advanced Engineering Informatics*, Vol. 26 No. 2, pp. 159-179. https://doi.org/10.1016/j.aei.2012.02.002.

Salvador, F. and Forza, C. (2004), "Configuring products to address the customization-responsiveness squeeze: A survey of management issues and opportunities", *International Journal of Production Economics*, Vol. 91 No. 3, pp. 273-291. https://doi.org/10.1016/j.ijpe.2003.09.003.

Sorri, K. et al. (2017), "*Prospects of CPQ: Evolving toward industry platforms*", CEUR Workshop Proceedings, pp. 3-15.

Stokes, M. (2001), "Managing engineering knowledge: MOKA: methodology for knowledge based engineering applications", *Strategy*, p. 257.

Sutherland, I.E. (1963), "*Sketchpad*", *Proceedings of the May 21-23, 1963, spring joint computer conference on - AFIPS '63 (Spring)*, ACM Press, New York, New York, USA, p. 329. https://doi.org/10.1145/1461551.1461591.

Tarkian, M. (2012), *Design automation for multidisciplinary optimization : a high level CAD template approach, [PhD Thesis]*, Department of Management and Engineering, Linköping University.

Trentin, A., Perin, E. and Forza, C. (2012), "Product configurator impact on product quality", *International Journal of Production Economics*, Vol. 135 No. 2, pp. 850-859. https://doi.org/10.1016/j.ijpe.2011.10.023.

Verhagen, W.J.C. et al. (2012), "A critical review of Knowledge-Based Engineering: An identification of research challenges", *Advanced Engineering Informatics*, Vol. 26 No. 1, pp. 5-15. https://doi.org/10.1016/j.aei.2011.06.004.

Wikner, J. and Rudberg, M. (2005), "Integrating production and engineering perspectives on the customer order decoupling point", *International Journal of Operations and Production Management*, Vol. 25 No. 7, pp. 623-641. https://doi.org/10.1108/01443570510605072.

Zhang, L. L. (2014) 'Product configuration: A review of the state-of-the-art and future research', *International Journal of Production Research*, Vol. 52 No. 21, pp. 6381-6398. https://doi.org/10.1080/00207543.2014.942012.