Reusing Components across Multiple Configurators

Amartya Ghosh[1] and Anna Myrodia[2] and Niels Henrik Mortensen[3] and Lars Hvam[4]

Abstract . The purpose of this paper is to examine the way in which an engineering company reuses components of existing configurators across multiple configurators. As the use of configurators has been extended across all lifecycle phases of products, product families, and services, companies tend to develop multiple configurators to support their business processes. Often, companies develop new configurators from scratch even though some existing configurators comprise components that serve a similar purpose. While the concept of reusability is discussed extensively in software and expert systems development literature, it has not been addressed in the existing literature on product configurators. In this study, the research team primarily focuses on the approach of reusing and sharing components from existing configurators to develop new configurators in a multi-configurator portfolio. We also examine the benefits and challenges of this reusability approach. The research is supplemented with empirical evidence based on an exploratory case study. The results demonstrate the way in which an engineering company uses and structures multiple configurators, the experiences with the concepts of reusing and sharing of configurator components and the lessons learned.

1 INTRODUCTION

The increased demand for highly customized product and service offerings has led to companies adopting mass customization strategies to reduce delivery times, lower costs, and to combat the challenges of product variant proliferation. This increase in product variation is accompanied by an increased amount of product information. This information is traded among the customers, the sales and the production departments at the company, and the suppliers to generate valid customized product variants and the requisite product documentation [1]. Companies use information technology (IT) tools such as product and service configurators to automate the handling of the product information [1].

Configurators are knowledge-based IT systems, which fulfill a configuration task. A configuration task is a special type of design activity [2] facilitated by a number of components, their corresponding properties and ports, and constraints which restrict the number of feasible combinations associated with the components [3]. Similarly, for service configurators, the configuration models for configurable services comprise types, their constituent attributes, and constraints organized in generalization and aggregation hierarchies [4].

The use of product configurators is associated with several benefits, that have both direct and indirect impact on the lead time, quality, and cost of the customizable products [1,5,6]. The literature reports these benefits in relation to the different lifecycle phases of complex configurable products [7], the impact on human resources and sales performance [8], the return on investment of a product configurator project [9] and the level of maturity of the company [10]. Examples from case studies demonstrate the quantitative value of these benefits [11,12].

On the other hand, several studies report a number of challenges that companies face in realizing the benefits from their projects. These challenges are categorized in relation to IT systems, product modeling, organizational issues, resource constraints, type of products, and knowledge acquisition [13]. Versioning control [14], ensuring data quality [15,16] and data maintenance [15] are some additional challenges that companies face while implementing a product configurator. A number of the previously mentioned challenges arise because of the high number of product variants, the complexity associated with the level of difficulty to model and maintain configurable products within a configurator, and the number of resources needed [17]. In the case of software and expert systems, the use of the concepts of modularisation and reusability has led to a reduction in development effort and risk and maintenance effort [18]. A number of studies on product configurators have addressed the issue of modeling a product family within a configurator, by incorporating the principles of product modularisation and product platform strategies into the underlying product model and developing system-level structures [19,20]. However, these studies do not address the implementation of the concept of reusability across multiple configurators or the benefits and challenges associated with the implementation of such an approach in an industrial setting.

Therefore, this study aims to address this gap in the existing literature by exploring the experiences of companies reusing or sharing components across multiple configurators. The practical implications of this study are examined via a case study of an engineering company using multiple configurators. The research investigates how the configuration team at the company reuses and shares components across multiple configurator projects and the benefits and challenges associated with the use of the concepts of reusability and sharing in the development of multiple configurators.

The structure of the article is as follows. Section 2 presents a literature review that provides a theoretical background on the reuse of parts or modules across multiple product architectures, general software systems, expert systems, and configurators. Section 3 presents the research method. Section 4 introduces the

[1] Department of Mechanical Engineering, Technical University of Denmark, Denmark, email: amgho@dtu.dk
[2] Department of Management Engineering, Technical University of Denmark, Denmark, email: annamyr@dtu.dk
[3] Department of Mechanical Engineering, Technical University of Denmark, Denmark, email: nhmo@dtu.dk
[4] Department of Management Engineering, Technical University of Denmark, Denmark, email: lahv@dtu.dk

case company and the landscape of its configurator portfolio, and in section 5, the results from the case study are presented. Section 6 discusses the results from the case study in relation to the research question and the existing literature base. Section 7 presents the conclusions to the study and the areas of future research.

2 THEORETICAL BACKGROUND

This section provides an overview of the existing literature on related topics such as the reuse and sharing of modules in software, product platform design and expert systems and identifies the gap in terms of reusability in configurators.

2.1 Modularity and reusability in products and product families

The concept of modularity has been discussed in depth in the recent literature in relation to the concepts of mass customization, product design, and complexity management [21]. The reason for the development and the extended advancement of the modularity concept, in both academia and industry, is related to the various benefits that its implementation is bringing along.

A module is defined as an "essential and self-contained functional unit relative to the product of which it is part" [22]. The standardized interfaces and interactions of each module enable the creation of product variants by developing and producing a combination of different modules. By pursuing a modularization strategy, companies can achieve economies of scale while offering greater product variety to their customers, increase strategic flexibility by reusing modules across various product models and model generations and concurrently develop the modules and product components [23]. In order to successfully implement a modularization strategy, the companies need to choose the suitable degree of modularity in their products, effectively prioritize the requirements of the company functions while designing the modules and coordinate the modular product development process across all the concerned organizational units [23].

The modularity of a product is an important characteristic of its architecture [24]. Based on the sharing of product architectures and standardized interfaces [25], the product platform approach enables companies to handle the proliferation of product variety. This approach entails the sharing of components, processes, knowledge, people, and relationships across a set of products [26].

The use of a platform strategy leads to a reduction in the development cost and time for new product variants. Moreover, the consequent reduction in the volume of parts and the number of associated processes leads to a reduction in material costs, logistics costs, procurement costs, inventory costs, and sales and services costs. [26]

However, companies face several challenges while adopting a product platform approach. Product planning and marketing managers have to decide on the product variants that will meet the demands of various market segments while saving development and production costs. Designers face challenges in deciding what product architectures to use in deriving product variants from product platforms. As more departments with differing goals and objectives get involved in the decision-making process, companies also face difficulties in maintaining a balance between the commonality and the distinctiveness of their products. [26]

2.2 Modularity and reusability in software

The reusability of software has been examined by academia in depth during the last few decades [27,28], regarding approaches and methods. With regards to engineering, there are three main concepts of reusability of software identified: application system, component and object and function reuse [29]. The entire application system can be integrated into other systems, without significant adaptations and changes to the reused system. An example of this reusability concept could be a commercial ERP system that is used by different companies with entirely different product portfolios. The component reusability refers to cases of reusing only a component of the whole system or sub-system to another. For instance, libraries in software can be reused by several systems containing information about different products. The last concept describes the reusability of software components that implement a single and well-defined object or function in a system. An example of this is the reusability of the price calculation logic described in a software component across several systems, such as quotation solutions, ERP systems, and order placing systems

As expected, there are several benefits associated with the reusability of software [18]. One of the main benefits is the ensured dependability of the software since the software has been already tested in another environment. Moreover, the reuse of exiting software leads to a reduction in development effort and the risk of implementation. For commercial software, the familiarity of the user with the system increases efficiency, productivity, and reduces the risk of errors. In particular, when reusing components or objects/functions of a system, it supports faster software development in terms of time, costs and resources, and it allows for specialists, who carry experience, knowledge and best practices regarding the reused modules, to be involved.

On the other hand, there are a number of challenges associated with the concept of software reusability in any of the three forms described before. The integration of a new piece of software into the existing IT landscape or the integration of an add-on modular component into an existing system usually requires customizations. These customizations can lead to compatibility issues. Furthermore, companies incur a high cost of hiring experts to make these customizations to the existing software.

Companies may also utilize a software-product line engineering (SPLE) approach to develop software systems by reusing assets created throughout the software product development lifecycle. This approach is characterized by two lifecycle processes: domain engineering and application engineering [30]. Domain engineering deals with the development of the reusable assets, which constitute the product line infrastructure [30]. Application engineering involves combining these reusable assets with product-specific assets to create the final software product [30]. Companies may combine several inter-dependent SPLs to create a multi-product line (MPL) to develop large or ultra-large software systems [31].

The benefits associated with the implementation of the SPLE approach include increased developer productivity, improved quality, reduced maintenance efforts, reduced code size, reduced consumption of resources, and reduced time-to-market of the products [32]. However, companies utilizing the MPL approach also face challenges in structuring the MPL models during the domain engineering process [33] and handling the technical and organizational dependencies between the constituent product lines during the derivation of a software product by multiple users [34].

2.3 Reusability in expert systems and product configurators

As with software systems, several studies have explored the use of the concept of reusability in developing expert systems [35]. An expert system refers to "a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice" [36]. It comprises a knowledge base, an inference engine, a knowledge acquisition system, and a user interface system [35]. At the highest level of abstraction, an expert system can be decomposed into two components: the knowledge base and the problem-solving methods, both of which can be reused to build other expert systems [35].

The creation of the knowledge base is a very resource-intensive task and is often a bottleneck in the development process of an expert system. Therefore, developers can achieve significant savings in time and costs by reusing the knowledge base across different problem-solving methods, even though the knowledge base may require adaptation to suit the problem scenarios. Developers can also reduce maintenance efforts by reusing previously tested problem-solving methods across multiple knowledge domains [35]. In certain studies, researchers have also addressed the decomposition of the problem-solving methods into constituent lower-level components in different environments, using architectures such as INDEX [37] and PROTÉGÉ [38]. Moreover, certain architectures and specifications such as CORBA [39] allow for the reuse of expert system components across different platforms and different development environments.

As configurators can generate valid configurations based on the underlying configuration model, they are considered to be typical examples of expert systems [40]. A couple of studies have proposed approaches for making system-level configurations.

An example of this approach is the SAP2 configurator that integrates sales, product and production configuration using specific sub-modules [19]. The configurator uses an underlying configuration model for a product family that unifies the functional view, the product component view and the corresponding production operations and resources view, called the GBoFMO (Generic Bill of Functions, Materials and Operations).

Another approach for system-level configuration identified in the literature is a configurator prototype that manages system-level platforms and incomplete product configurations for engineer-to-order (ETO) companies and project-based businesses [20]. These projects consist of system-level configurations and multiple product configurations. Each product configuration is decomposed further into its constituent subsystems and parts. The system-level configurations are based on high-level templates, containing the system level parameters. The system configuration provides inputs into the product configurations, while the product configurations feedback any changes to the parameters to the system-level configuration. The system-level configuration instantiates the product configurations through the use of a common template and domain-specific vocabulary.

2.4 Benefits and challenges of reusability

In the previously mentioned studies on the challenges of configurator implementation and usage, a number of companies were unable to realize the benefits from the implementation of configurators. The reasons for this inability include difficulties in acquiring, formalizing and managing product knowledge, handling rapid product changes and the inability of the configurators to cover the entire product portfolio [41]. Companies producing highly complex products also faced challenges in clearly defining the product families that would be represented in the configurators [13]. Moreover, a number of companies faced challenges due to a lack of resources for developing and maintaining the configurators and developing integrations to other IT systems [42]. The companies can reduce the development and maintenance efforts for their portfolio of configurators by adopting the concepts of reusability and sharing, as utilized in the design of product platforms, general software systems, and expert systems, in the development of new configurators. However, the aforementioned studies on the SAP2 configurator and the system-level configurator do not explicitly address how companies can adopt the concepts of reusability and sharing in structuring multiple configurators and the benefits and challenges they face in the implementation of such an approach.

Thus, there is a lack of empirical evidence into how companies, which have successfully implemented multiple configurators, structure their configurators, and what components are reused or shared across different configurators. This study aims to address this gap by conducting a case study at a company using multiple configurators to find out how companies utilize the concept of reusability in structuring their configurators and what benefits and challenges they face by adopting this particular approach. To achieve this goal, the following research question is formulated:

RQ: *How do engineering companies utilize the concept of reusability and sharing to structure their product configurators?*

3 RESEARCH METHOD

To answer the RQ, the research team conducted an exploratory case study at a Danish engineering company utilizing the concept of reusability and sharing of components across multiple configurators to support the sales processes. The specific company was selected as it is considered to be representative in terms of using multiple product and service configurators to support the sales processes pertaining to their ETO products. The company is sufficiently mature as it has been using configurators for sixteen years. The reason for applying case study research is to test still unknown variables and not entirely understood phenomena in their natural settings [43]. The unit of analysis in this study is the configurator portfolio which has been built up by the company.

Data collection was conducted in the form of semi-structured interviews with members of the configuration team at the company. The interviewees were selected based on their years of experience, knowledge, and level of involvement in designing the configurator set up. The research team interviewed the IT project manager and the business owner from the configuration team as both these team members had extensive experience in the development, implementation and maintenance of the configurators. The business owner is responsible for coordinating with the stakeholders from the different business areas and in the prioritization of new configurator projects. She has been working on configurators at the company for the last seven years. The IT project manager was responsible for the project management tasks related to the configurators and supported the team in handling some maintenance tasks on the configurators. She has been working with the configuration team for nearly four years.

The research team opted to conduct semi-structured interviews to impose an overall structure to the ensuing discussion and to provide a direction to guide the interviewees while allowing for some flexibility for the interviewees. Based on the reviewed literature on the concepts of modularity and reusability in physical products, software systems and expert systems discussed in the theoretical background section, the research team prepared a list of questions, which are presented in the Appendix. The duration of each interview was one hour.

The questions address the topic of reusing and sharing configurator components across multiple configurators and the benefits and challenges that the company has faced while implementing such an approach. The questions, categorized as "Company-specific questions", aim at providing an overview of the case company's experience in developing and implementing configurators. The interviewee specific questions address how the configuration team has set up multiple configurators and how they have utilized the concept of reusing and sharing configurator components across multiple configurators.

As the research team was also investigating the aforementioned approach from a business perspective, the last two interviewee-specific questions address the impact of implementing this approach. In particular, the interviewees were asked about the benefits and the challenges of implementing a modular concept on the product configurator portfolio. First, they were asked to answer this question based on their experience and then they were presented with a list from the benefits and challenges identified in the literature, to ensure that all the relevant topics were taken into account.

The collected data used for the analysis were qualitative in nature, primarily based on the interviews. Supplementary material provided included schematics of the IT landscape and a demonstration of the configurators, to allow the research team to develop a better understanding of the configurator portfolio.

4 CASE DESCRIPTION

The selected case company is an engineering company, providing solutions from specific equipment to complete plant solutions. The company operates globally, serving the food, dairy, chemical, and pharmaceutical industries, and has a yearly turnover of approximately 305 million €. The complete solutions provided include some standardized customize-to-order (CTO) products, but they also require customized ETO products for specific customers. As mentioned earlier, the company has been using configurators for sixteen years, with the current portfolio of configurators covering approximately 50% of the entire product portfolio of the company. In the case of ETO products, the concerned configurators are capable of generating full or partial configurations.

The configurators support the sales and service phases of the products and solutions offered to the customers, particularly the tendering and procurement processes. The configuration team consists of 5 people in total: the business owner, an IT project manager, one configuration engineer focusing on product and service modeling activities, and two software developers. The software developers focus primarily on developing add-ons and plug-ins to enhance the functionality of the configurators. They are also responsible for the development and maintenance of the integrations with the other IT systems being used by the company. The team is supported by a super-user from each business area, who is responsible for collating change requests from the end-users of the configurators and documenting these requests in a dedicated documentation system. Moreover, the configuration team is also supported in the development and maintenance of a number of configurators in a specific business area by another configuration team. Only the configuration teams are allowed to make updates and perform maintenance tasks on the configurators.

4.1 IT landscape: Configurators and integrations

The overall set-up of the configurators at the company can be described in four levels. Figure 1 illustrates the distinction between the different levels. The plant configurator describes the complete plant, including all the specific equipment and the services related to them. The plant configurator also contains some constraints and knowledge about the specific plant type covered by the configurator. Each plant is first decomposed into several systems, specific to each plant. Each system-level configurator contains knowledge about the variants of each system. On the equipment level, each configurator contains product knowledge and constraints pertaining to specific equipment. The company also uses a global service configurator, which is used across the equipment configurators.

When configuring a plant, the plant configurator calls the specific system configurators, which, in turn, calls the requisite equipment level configurators. The service configurator provides service information for different plants and equipment.
All the configurators are developed using the same commercial configuration system software.

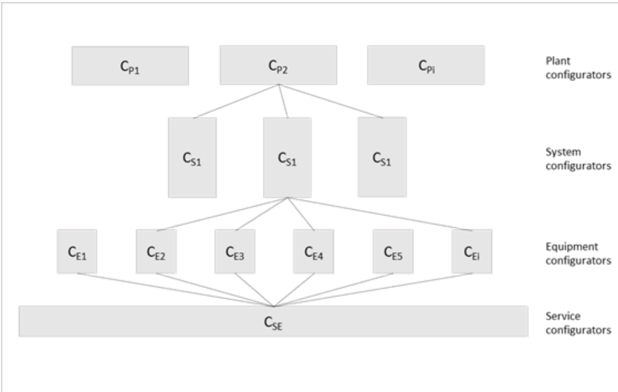

Figure 1. Multi-level configurator set up at the case company

At the plant level, the plant configurators have integrations to the internal software system for making engineering calculations, the pricing and the quotation databases, the product data management system for the generation of piping and instrumentation drawings, the document generator system and the calculation portal that is used for manual cost calculations and importing the values to the ERP system for project budgeting purposes. At the equipment level, the configurators are only integrated to the document generator system.

Apart from the integrations to external systems, the commercial configuration system software that the company uses enables each configurator to call other configurators at a lower level. From a practical point of view, to configure a complete solution (plant), the configuration process starts from the plant level. At this level, the user decides on the plant systems and based on this selection, the required equipment are then individually configured. However, in certain scenarios, the end-users can also use the equipment configurators independently, without having to configure any overarching plants or systems. For example, end-users in the procurement department might require the prices for a specific configuration of particular equipment and therefore would only need to use an equipment configurator instead of creating a plant-level configuration and a plant system-level configuration first.

A super-user from each business unit is responsible for the collation of change requests or bugs from the end-users of the particular configurators under the purview of the business unit. The super-users document their requests in a dedicated documentation system, which the configuration team can use while updating the concerned configurator models. After making a change to the particular configurator model, the configuration team follows a standard procedure to document the changes. Then, the IT project manager approves the changes before they are released for use by the business units. The configuration team also uses a version control system for the configurator models allowing them to revert to an older version of the configurator model in case any issues arise from the changes made.

5 RESULTS

The following section presents and discusses the results of the case study. It focuses on the implementation of the concept of reuse and sharing of configurator components at the case company and the benefits and the challenges the company has experienced using this approach.

5.1 Reusability and sharing

The configurators have a high degree of sharing and reusing parts of the system, both within the same level and across different levels, as illustrated in Fig. 1. When reusing components from existing configurators to develop new configurators, the configuration team uses the product model from an existing configurator, either entirely or partially, and adapts it to suit the purposes of the new configurator project. For instance, two equipment in two separate business application environments might have product models which are similar, but one equipment may be more complex than the other. In this case, the configuration engineer may reuse the existing product model of one equipment, either partially (by making some changes) or fully while developing the configurator for the other equipment.

Additional areas of component reuse across different configurators at the same level relate to the reuse of logic components, e.g. ways of calculating values and generating documents. The components of the configurator model that are reused are usually the ones that do not require frequent changes to the structure of the product model.

At the plant system level, the concept of sharing becomes more apparent. As mentioned earlier, each plant is decomposed into a number of systems, which in turn, are composed of certain equipment. These plant-systems are unique to each plant. However, the same type of equipment may be used to meet the needs of different plant-systems and plants. In such cases, the equipment level configurators are shared across multiple plant system configurators at the plant system level. For example, if a family of blowers can be used in two different plants, then the company only uses one configurator to store the product knowledge, and the two different plant configurators call that equipment configurator when required.

As the plant level configuration is highly dependent on the business application environment, the responsible business units are primarily responsible for deciding what can be shared, for example, based on the material used for the different equipment and the prices. The business units are responsible for the overall setup of the plants and the constituent plant-systems and equipment. When a particular business unit requests the development of a configurator for new equipment, the configuration team starts building a configurator model based on the input from that business unit. However, based on the tacit knowledge of the configuration team and the documentation available on the existing configurator models, the team may decide to reuse an existing model in another application environment and adapt it to meet the needs of the current context. Furthermore, the business units may also coordinate with the procurement department to find any existing configurators which may already cover their user requirements.

5.2 Benefits

The primary benefit reported during the interviews was the standardization achieved across the business units in the company. Standardization is a benefit that is usually associated with configurators. In this case, the standardization refers to both the products and the processes. The standardization covers not only the product models that were modeled into the configurators but also the sales processes supported by them, the roles and responsibilities of the stakeholders, and the maintenance and change tasks associated with the configurators. Having these standard procedures in place allow the configuration team and the business units to improve their efficiency, to improve communication and keep track of the change requests and changes to the configurator models.

Furthermore, the multi-configurator set-up allows for a more modular representation of the product portfolio, including both individual products but also complete solutions. In particular, the introduction of the plant-system configurators supported the decomposition of the plant into its constituent systems, which resulted in rendering the development, maintenance, and testing tasks much easier and quicker.

The interviewees assigned more importance to the benefits of standardization of the associated processes as compared to the reduction in the development, maintenance, and testing efforts. Nevertheless, there are variations identified regarding the time spent of these tasks. In particular, the development time for an equipment configurator is approximately six months along with an additional one or two months allocated for initial testing, whereas the plant configurators take around one year to develop. During the development phase of a configurator project, the effort spent in knowledge acquisition is more significant compared to the actual effort dedicated to the modeling of the configurator.

With reference to handling changes, the interviewees again highlighted the improvements in efficiency. These improvements arise due to the differences in frequencies of change requests across different levels. For instance, the plant configurator models were quite stable and required one or two changes annually. Since the frequent changes are generally limited to the equipment configurators, the maintenance and testing effort is lower as compared to the effort required for maintaining and testing the plant configurators. By having a clear overview of the tasks required and to what level of the configurator portfolio they are assigned to, the team can predict the maintenance efforts more accurately and improve the efficiency of executing these tasks

Another benefit highlighted is related to improved communication and knowledge sharing in the company. The particular setup of the configurator portfolio allows for better communication among various stakeholders, especially cross-organizational, e.g. when a business unit raises a request for new configurators with the configuration team. In such a scenario, the configuration team can use existing configurators for similar equipment or plants within the same or different application environment to show the scope and the functionalities, which the configurators offer, to the stakeholders from the business unit. In that sense, the reusability refers both to the configuration models and the knowledge and experience for the processes supporting the configurator.

5.3 Challenges

On the other hand, the interviews revealed several challenges that the case company is facing while implementing the concepts of sharing and reusability across the different configurators. The main challenge concerned the coordination of the teams. As mentioned earlier, two configuration teams are responsible for developing, implementing and maintaining the configurators and their integrations to other IT systems. If an equipment configurator model is updated without informing the rest of the team, then this might adversely affect the functioning of the overarching plant models containing that equipment configurator. The interviewees emphasized the coordination between the teams to ensure that such issues would not arise while making changes or updates to existing models. This is because the case company operates globally and the geographical location differs among the team members. The coordination related challenges are also in terms of roles and responsibilities. Even though the set-up of the configurators ended up supporting the transparency of the roles and tasks, in the beginning, it was not clear how the distinction was made and how the changes to the configurator models were communicated to the stakeholders.

Version control is another important challenge faced by the company. While the configuration team utilizes a versioning control system for the entire set of configurators, the team still faces a challenge in deciding which version of the product model to save. When the changes to any of the configurators are limited, (e.g. update of prices), the configuration team does not save the previous version first before making the changes. However, when the change affects several levels of the configurator set up, the team requires more time for implementation and testing. In this case, the previous version is saved as it is significantly different from the updated version. Another challenge faced by the configuration team is the maintenance of compatibility across the plant configurators, the plant-system configurators and equipment configurators. In certain situations, the changes made to the plant configurators result in errors if the lower-level models are not updated accordingly.

The company also faces challenges in deciding on the role of the business units in the scoping and decision-making phases of the configurator development projects. The business units are responsible for making decisions regarding the overall structure of the configurators. They also play a crucial role in defining the parts of the configurators that are shared based on their alignment with the strategic goals of the company and the market needs. The initial user requirements, that are set from the business units, always need to be adjusted, requiring several iterations and quite often the final result is very different from the initial one.

6 DISCUSSION

This study addresses the issue of how engineering companies using multiple configurators utilize the concept of reusability and sharing of configurator components in developing their configurators, and the benefits and challenges that the companies face in adopting this approach. The study presents the case of a Danish engineering company, which has been using configurators to configure ETO plants and the constituent plant-systems and equipment.

The findings from the literature study indicate that the concepts of reusability and sharing have been extensively examined in the fields of product development, software systems, and expert systems. In the case of products, the use of reusable modules and product platform strategies benefits companies by leading to a reduction in development time, cost of new product variants, manufacturing costs, and inventory costs. In case of software systems and expert systems, the reuse of components, such as libraries, problem-solving methods, and knowledge bases, lead to the reduction in development, testing and maintenance effort associated with these systems. These findings from the literature study, along with the results from the case study, provide an answer to the RQ. The empirical evidence demonstrates how an ETO company develops multiple configurators to support its business processes by reusing and sharing different configurators and configurator components. It also explains how these concepts are utilized when developing new configurators for different business units of the company. Furthermore, the study also addresses the benefits and challenges associated with the implementation of such an approach.

The way in which the case company structures their configurators into different levels is similar to the system-level configuration approach proposed in [20]. In both the approaches, the system-level (plant-level, in case of the case company) configurator is modeled first, followed by the modeling of the constituent configurations. Both approaches also allow for partial configurations for ETO products. However, the system-level configurator prototype, as described in [20], does not address the issue of using the concept of reusability and sharing of configurator components across multiple configurators covering different projects or ETO products and its benefits and challenges. The benefits that the case company has generated from the use of the concept of the reusability and sharing of configurator components are similar to the benefits of the reuse of knowledge bases and problem-solving methods in expert systems [1,5]. However, the interviewees noted that the development time of the model itself was insignificant compared to the time required for product knowledge acquisition from the business units.

With reference to the challenges of implementing a modular approach to the set-up of the configurator portfolio, the findings from the case study are aligned with the findings from the literature [13,14]. Change management, knowledge acquisition and maintenance of the models are the main issues addressed in the findings of this research and can be primarily associated with the concepts of sharing and reusing parts of the configurators.

These findings provide strong empirical evidence to support managerial decisions in terms of designing and structuring a configurator portfolio. The insights from the case study can be used as guidance when defining the scope of a multi-level configurator set up and the establishment of the cross-organizational collaboration among the teams involved in the development process.

7 CONCLUSION AND FUTURE RESEARCH

The focus of this study is on examining the concepts of reusability and sharing across multiple configurators. While these concepts are well established in the fields of product modeling and software development, they have not been addressed in depth in relation to product configurators. The research team conducts a case study to investigate how these concepts are implemented in an industrial setting. The results from the literature review and the case study provide an answer to the developed RQ on how the concept of reusability in configurators is being used by engineering companies.

This study contributes to the existing knowledge on the modeling and scoping of configurators by looking at how an engineering company structures multiple configurators for configuring ETO products using the concepts of reusability and sharing of different configurator components. Practitioners in the industry can also gain some insights into adopting these concepts while modeling and scoping product configurator projects, as the case company is representative of engineering companies producing complex ETO products and utilizing multiple configurators to support their business processes.

This study focuses only on one case company and how they use the concept of reusability and sharing of configurator components to structure multiple configurators. Consequently, the discussion of the benefits and the challenges, which arise out of the usage of this concept, pertain only to the specific case company and the scope and structure of their configurators. Therefore, future work will focus on increasing the number of case companies, thereby making the results of the study more generalizable to the industry at large.

Another limitation of the study relates to the role of the interviewees. For this study, we have interviewed only the business manager and IT project manager on the company's configuration team. However, the business and technical units are responsible for the underlying product architectures that are modeled in the configurators. Therefore, in future work, we aim to interview business unit stakeholders at the case companies to incorporate their insights into the way in which the concept of reusability and sharing is implemented and the benefits and challenges that they perceive arising out of the implementation of this approach in developing configurators.

REFERENCES

[1] Forza C, Salvador F. Managing for variety in the order acquisition and fulfilment process: The contribution of product configuration systems. Int J Prod Econ 2002;76:87–98. doi:10.1016/S0925-5273(01)00157-8.

[2] Mittal S, Frayman F. Towards a generic model of configuration tasks. Ijcai-89 Proc Elev Int Jt Conf Artif Intell 1989:1395–401 vol.2.

[3] Felfernig A, Friedrich GE, Jannach D. UML as domain specific language for the construction of knowledge-based configuration systems. Int J Softw Eng Knowl Eng 2000;10:449–69.

[4] Heiskala M, Tiihonen J, Soininen T. A conceptual model for configurable services. Pap. from Config. Work. IJCAI'05, 2005, p. 19.

[5] Hvam L, Haug A, Mortensen NH, Thuesen C. Observed benefits from product configuration systems. Int J Ind Eng Appl Pract 2013;20:329–38.

[6] Felfernig A, Hotz L, Bagley C, Tiihonen J. Knowledge-based configuration: From research to business cases. Newnes.; 2014.

[7] Myrodia A, Kristjansdottir K, Shafiee S, Hvam L. Product configuration system and its impact on product's life cycle complexity. IEEE Int. Conf. Ind. Eng. Eng. Manag., vol. 2016-Decem, 2016. doi:10.1109/IEEM.2016.7797960.

[8] Haug A, Shafiee S, Hvam L. The costs and benefits of product configuration projects in engineer-to-order companies. Comput Ind 2019;105:133–42. doi:10.1016/j.compind.2018.11.005.

[9] Kristjansdottir K, Shafiee S, Hvam L, Bonev M, Myrodia A. Return on investment from the use of product configuration systems – A case study. Comput Ind 2018;100:57–69. doi:10.1016/j.compind.2018.04.003.

[10] Myrodia A, Randrup T, Hvam L. Configuration lifecycle management maturity model. Comput Ind 2019;106:30–47. doi:10.1016/j.compind.2018.12.006.

[11] Hvam L. Mass Customization in the electronics industry. Int J Mass Cust 2006;1:410–26.

[12] Forza C, Salvador F. Product configuration and inter-firm co-ordination: an innovative solution from a small manufacturing enterprise. Comput Ind 2002;49:37–41.

[13] Kristjansdottir K, Shafiee S, Hvam L, Forza C, Mortensen NH. The main challenges for manufacturing companies in implementing and utilizing configurators. Comput Ind 2018:196–211. doi:10.1016/j.compind.2018.05.001.

[14] Heiskala M, Tihonen J, Paloheimo K-S, Soininen T. Mass Customization with Configurable Products and Configurators. Mass Cust. Pers. Commun. Environ., IGI Global; 2011, p. 75–106. doi:10.4018/978-1-60566-260-2.ch006.

[15] Rasmussen JB, Myrodia A, Mortensen NH. Cost of Not Maintaining a Product Configuration System. Int J Ind Eng Manag 2018;9:205–14. doi:10.24867/IJIEM-2018-4-205.

[16] Haug A, Zachariassen F, Liempd D van. The costs of poor data quality. J Ind Eng Manag 2011;4:168–93. doi:10.3926/jiem.2011.v4n2.p168-193.

[17] Asadi M, Soltani S, Gašević D, Hatala M, Bagheri E, Benavides D, et al. The effects of visualization and interaction techniques on feature model configuration. Empir Softw Eng 2016;21:1706–43. doi:10.1007/s10664-014-9353-5.

[18] Williams LG, Smith CU. PASASM: A Method for the Performance Assessment of Software Architectures. In: Balsamo

S, Inverardi P, Selic B, editors. WOSP ’02 Proc. 3rd Int. Work. Softw. Perform., Rome, Italy: ACM New York, NY, USA; 2002, p. 179–89. doi:10.1145/584369.584397.

[19] Zhang LL, Vareilles E, Aldanondo M. Generic bill of functions, materials, and operations for SAP2 configuration. Int J Prod Res 2013;51:465–78. doi:10.1080/00207543.2011.652745.

[20] Kristianto Y, Helo P, Jiao RJ. A system level product configurator for engineer-to-order supply chains. Comput Ind 2015;72:82–91. doi:10.1016/j.compind.2015.04.004.

[21] Suzić N, Forza C, Trentin A, Anišić Z, Suzi N, Ani si Z. Implementation guidelines for mass customization: current characteristics and suggestions for improvement. Prod Plan Control Manag Oper 2018;29:856–71. doi:10.1080/09537287.2018.1485983.

[22] Miller TD, Elgard P. Defining modules, modularity and modularization. Proc. 13th IPS Res. Semin. Fuglsoe, 1998.

[23] Persson M, Åhlström P. Managerial issues in modularising complex products. Technovation 2006;26:1201–9. doi:10.1016/j.technovation.2005.09.020.

[24] Ulrich K, Eppinger SD. Product Design and Development. New York: McGraw-Hill; 1995.

[25] Harlou U. Developing product families based on architectures: Contribution to a theory of product families. Technical University of Denmark (DTU), 2006.

[26] Robertson D, Ulrich K. Planning for Product Platforms. Sloan Manage Rev 1998;39:19–31.

[27] Raatikainen M, Tiihonen J, Männistö T. Software product lines and variability modeling: A tertiary study. J Syst Softw 2018;149:485–510. doi:10.1016/j.jss.2018.12.027.

[28] Mohabbati B, Hatala M, Gašević D, Asadi M, Bošković M. Development and Configuration of Service-Oriented Systems Families. Proc. 2011 ACM Symp. Appl. Comput. (SAC ’11), TaiChung, Taiwan: ACM; 2011, p. 1606–13. doi:10.1145/1982185.1982522.

[29] Singh S, Singh S, Singh G. Reusability of the Software. Int J Comput Appl 2010;7:38–41.

[30] Pohl K, Böckle G, Van Der Linden F. Software product line engineering: Foundations, principles, and techniques. Springer Berlin Heidelberg; 2005. doi:10.1007/3-540-28901-1.

[31] Holl G, Grünbacher P, Rabiser R. A systematic review and an expert survey on capabilities supporting multi product lines. Inf Softw Technol 2012;54:828–52.

[32] Van Der Linden F, Schmid K, Rommes E. Software product lines in action: The best industrial practice in product line engineering. Springer Berlin Heidelberg; 2007. doi:10.1007/978-3-540-71437-8.

[33] Rabiser R, Grünbacher P, Holl G. Improving awareness during product derivation in multi-user multi product line environments. CEUR Workshop Proc 2010;688:1–5.

[34] Holl G, Vierhauser M, Heider W, Grünbacher P, Rabiser R. Product line bundles for tool support in multi product lines. Proc. 5th Work. Var. Model. Software-Intensive Syst., 2011, p. 21–7.

[35] Gennari JH, Stein AR, Musen MA. Reuse for knowledge-based systems and CORBA components. Proc 10th Banff Knowl Acquis Knowledge-Based Syst Work 1996:461–4616.

[36] Jackson P. Introduction to Expert Systems. 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.; 1998.

[37] Dai W. Reusable expert system components development. Appl Artif Intell 1996;10:225–38. doi:10.1080/088395196118560.

[38] Gennari JH, Musen MA, Fergerson RW, Grosso WE, Crubézy M, Eriksson H, et al. The evolution of Protégé: an environment for knowledge-based systems development. Int J Hum Comput Stud 2003;58:89–123.

[39] Gennari JH, Cheng H, Altman RB, Musen MA. Reuse, CORBA, and knowledge-based systems. Int J Hum Comput Stud 1998;49:523–46.

[40] Hvam L, Riis J, Mortensen NH. Product customization. Berlin Heidelberg: Springer; 2008.

[41] He W, Mingz XG, Nix QF, Luy WF, Lee BH. A unified product structure management for enterprise business process integration throughout the product lifecycle. Int J Prod Res 2006;44:1757–76. doi:10.1080/00207540500445453.

[42] Colombo G, Furini F, Rossoni M. The Role of Knowledge Based Engineering in Product Configuration. In: Eynard B, Nigrelli V, Oliveri S, Peris-Fajarnes G, Rizzuti S, editors. Lect. Notes Mech. Eng., Cham: Springer; 2017, p. 1141–8. doi:10.1007/978-3-319-45781-9_114.

[43] Voss C, Tsikriktsis N, Frohlich M. Case research in operations management. Int J Oper Prod Manag 2002;22:198–219. doi:10.1108/01443570210414329.

Appendix

Company specific questions

1. How long has the company been using configurators?
2. What is the scope of the configurators?
3. Which areas of the product lifecycle do the configurators support?
4. To what extent do the configurators cover the product and services portfolio?
5. What is the nature of the products that are covered by the configurators?
6. What is the setup of the configuration team at the company?

Interviewee specific questions

1. How is the configurator portfolio set up?
2. What components do configurators share/reuse within/across the configurator portfolio? How do the configurators share/reuse these components?
3. How does the configuration team decide on what components to share?
4. How do you visualize/represent your product configurator models and how do you introduce new configurators into the existing portfolio/architecture?
5. What are the benefts and challenges of implementing this approach?
6. How has the planning, development and implementation of configurators changed over the years at the company? What are the reasons behind this change?