Enterprise Modelling and Information Systems Architectures
Vol. 13, No. 16 (2018). DOI:10.18417/emisa.13.16

Special Issue on Smart Service Engineering

Product-Service-Systems

What and why Developers can learn from Mass Customization

Paul Christoph Gembarski*,[a], Roland Lachmayer[a]

[a] Institute of Product Development, Leibniz University Hannover, Welfengarten 1A, 30167 Hannover, Germany

Abstract. *Despite their very similar objectives, delimitations or associations between the two business types of mass customization and providers of product-service-systems (PSS) cannot be found in literature. In the following article, both business types are compared with each other and mapped into a common business-typological framework, the product-process-baseline-change matrix. Following that, the development of PSS is characterized especially with regard to the (re-)configurability of PSS over the life-cycle. Since product configuration is one of the key tools in the development and the customer co-design process in mass customization, its application to PSS is evaluated and present PSS-configuration approaches are discussed.*

Keywords. Product-Service-System Configuration • Mass Customization • Solution Space Modeling

Communicated by O. Thomas. Received 2017-03-10. Accepted after 2 revisions on 2018-06-10.

1 Introduction

The suppliers of technical products in many industrial sectors are exposed to increased competitive pressure. On the one hand, many markets are becoming progressively heterogeneous, which is reflected by a wide range of customer needs and leads to an ever more differentiated offering. On the other hand, the constraints are globalization of supply and demand. Addressing the complexity of order acquisition, product development and finally manufacturing and distribution are critical factors of success (Herlyn 2012).

In this context, economic success is also determined by the methods of variant design, like platform strategy and modular design kits, which especially are used in the automotive industry (Renner 2007).

The apparent contradiction between the diversity of the product offers on the one hand and stable as well as efficient mass-production processes on the other hand is resolved particularly in mass customization business models (in the following referred to as MC). The companies that operate MC aim to generate highly specialized, tailor-made solutions by integrating the customer into a co-design process in order to achieve enduring and sustained customer loyalty (Reichwald and Piller 2009).

Similar objectives are pursued by the suppliers of product-service-systems (hereinafter referred to as PSS) which main characteristic is the integrated and coequal development of product and service components (Gräßle et al. 2010). In particular, developers of such a PSS aim at not capturing customer requirements solely prior to and during product development, and to implement them in a single technical solution, but rather to accompany customers throughout the entire product life-cycle. So, through the exchange or reconfiguration of product and service components, the PSS supplier may react to new or changed requirements (Müller 2013).

Despite the very similar objectives, literature rarely shows any delimitations or connections between the two business types. Comparisons are limited to the analysis of the value enrichment

---

* Corresponding author.
E-mail. gembarski@ipeg.uni-hannover.de

Note: This work is based on Gembarski and Lachmayer (2016).

capabilities of individually configured products in contrast to standard products and the possibility of customer integration (Vogel-Heuser et al. 2014). The analysis of the product development processes and tools for MC as well as their assessment with respect to their application for PSS development is still in progress. This article bridges a part of this gap.

Thus, we raise the questions: (1) May MC and PSS be mapped in a common business typological framework? And (2) based on the hypothesis that MC methods and tools are also suitable for PSS, what kind of computer-aided configuration tools have been implemented for PSS already and what is the resulting research gap with respect to configuration of MC offerings?

In order to answer these questions, the remainder of this article is organized as follows: In Sect. 2, we provide a literature based comparison of the two business types mass customizer and supplier of PSS. Both are then integrated into the product-process-baseline-change matrix as business typological framework. Afterwards in Sect. 3, the development and configuration of PSS-artifacts is examined. Sect. 4 then provides an overview of solution space modeling focused on configuration systems and classifies existing PSS configuration approaches. Closing the article, Sect. 5 contains concluding remarks and drafts further research questions. Although mainly conceptual, the article targets researchers, practitioners and students with an interest in state-of-the-art conceptual and enterprise modeling research and its applications as well as in computer-aided engineering of PSS constituents.

2 Analysis of Business Types

The concept of mass customization was introduced by Davis into the scientific discussion at the end of the 1980s (Davis 1987) and further characterized by Boynton et al. as the ability to offer individualized products manufactured and distributed with mass production efficiency (Boynton et al. 1993).

In order to get a fundamental understanding of MC, the following section is used to derive the related business type from the product-process-change matrix. This is followed by a description of the characteristics of mass customizers. Subsequently, PSS are characterized and integrated into the business typology framework. The concept of business typology is used within this paper in the sense of Miles and Snow, who classify companies based on the combination of market strategy, organizational structure, company-internal processes and management theory (Miles and Snow 1986).

Looking at the individual business models, i. e. the model of the relationships how benefits for customers or other actors in the value chain are generated and returned to the company in form of turnovers (Schallmo 2013) is beyond the scope of this article.

2.1 Product-Process-Change Matrix

The framework described by Boynton et al. (Fig. 1) is a theoretical-deductive business typology from which four fundamental competitive strategies and their corresponding transformation rules are derived. The basis is a model for corporate change. In Fig. 1, product change is listed on the one axis as a unit for new products or product variants, while process change includes all process steps and technologies to develop, produce and market these new products (Boynton et al. 1993).

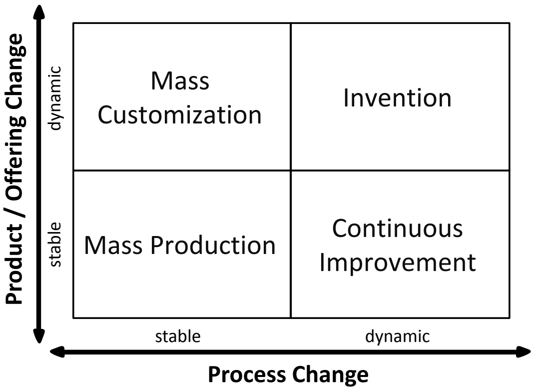

Fig. 1: Product-Process-Change Matrix (Boynton et al. 1993)

Both forms of change may occur either stably or dynamically. The first means that change takes place slowly and foreseeably, while the latter is fast, revolutionary and generally unpredictable.

Invention refers to a differentiation-based job shop production, in which new products and the respective processes for their development and manufacturing are constantly designed and newly generated. Products which are suitable for mass marketing are further developed to *Mass Production* goods. Here, scale effects have to be exhausted as far as possible, which in turn means that the production process must be kept stable. Any disturbance (either by product adaptation or a new production variant) leads either to increasing set-up costs or unwanted start-up effects (learning curve, increased waste rate, etc.). Boynton et al. point out that there is a synergy between the two types of invention and mass production. The latter is not capable of producing completely new and innovative products from itself, thus it must be served from the former.

The so-called *Continuous Improvement*, which usually operates in highly segmented or niche markets, is introduced as third type of competitive strategy. This type usually follows mass production and focuses on rationalization, process and quality improvement. Known approaches for this are TQM or Kaizen (Reichwald and Piller 2009). These measures are accompanied by a steady expansion of the product portfolio and the occupying of market niches.

Mass Customization is the fourth business type and focuses on Pine's so-called dynamic stability (Pine and Davis 1993). This means that products can be tailor-made to customer specifications (especially in delimitation to *Invention*) by the use of flexible yet stable processes in product development and production. Important principles for achieving this and mastering the resulting product complexity are e. g. product configuration and modular design kits (Gembarski and Lachmayer 2015a). On the production side, the technologies of additive manufacturing are considered as enabler (Lachmayer et al. 2015).

2.2 Characteristics of Mass Customizers

As a competitive strategy, MC focuses on the possibility of customization and individualization by the customer, and on the other hand on the application of flexible goods and service creation processes with mass production efficiency.

The first results in a consistent focus on the customer, since only he is able to formulate his specific needs and requirements for a good or service. Here, Pine et al. introduce the strategic aspect of a so-called learning relationship. In this case, the customer enables the supplier to recognize his or her needs over time and to anticipate them if necessary so that the customer is always served with his demands and needs. Once such a relationship has been established, switching to another vendor is associated with relatively high transaction costs, and customer loyalty increases (Pine et al. 1995). Piller identifies as a central element of this interaction a customer centric co-design process for both - products and (accompanying) services - to meet the individual needs (Piller 2004).

By emphasizing "mass" and the associated product development methods as well as manufacturing technologies, MC is delimited from traditional job shop production. In order to overcome the apparent contradiction between individual products and mass production, it is necessary to carry out the co-design process within a defined, stable solution space, which is designed both for the precise specification of customer requirements as well as according to the fast reaction of the production and distribution networks. This results in various individualization strategies (and, in turn, different business models) which differ in the degree of customer integration and the resulting influence on the value chain (Gembarski and Lachmayer 2015b).

Examples are *set-up customization* and *aesthetic co-design*. The first is understood by the authors as the possibilities of product adaptation, especially in the case of mechatronic devices, that result from the parametrization of their software component. Examples of this are combustion engines whose acceleration and consumption characteristics can be managed by the corresponding engine control units, or the variety of mobile applications which, although largely composed of identical physical components, can be configured differently in their functionality by the recorded software apps. The influence of the customer on the development and production of hardware components is extremely low in this business model. By contrast, aesthetic co-design aims at the customer's involvement by allowing him to vary the appearance of the product according to his or her own requirements. This refers firstly to color and texture, but also to its shape. The design space must be defined in advance by the supplier in such a way that, on the one hand, there is no impairment of the final product (e. g. because a housing has been modeled too small and collides with other components or a design interface between housing and module carrier has been changed so that final assembly is no longer possible). On the other hand, it must be ensured that the customer variants can be produced efficiently with existing production facilities (Gembarski and Lachmayer 2017).

Pine et al. also state that the company's own processes, whether administrative or directly related to the value creation, are to be developed as a modular system that is specifically configured for a customer solution as required. In the broadest sense, this also relates to the compilation of the supply network. Its coordination is often centralized, while the individual modules have organizational responsibility for their processes and results (Pine et al. 1993).

The key characteristics of a mass customizer are summarized in Tab. 1. For further features and a discussion of the success factors for MC, see Da Silveira et al., Dabic and Fogliatto et al. respectively (Da Silveira et al. 2001, Dabic 2006, Fogliatto et al. 2012). There is also an overview of successful MC implementations in the capital goods industry, the telecommunication sector, in the food and beverage industry, as well as in clothing and footwear. Further examples of insurance, financial institutions and other service providers are presented in Boynton et al. (1993) and Pine et al. (1995).

Tab. 1: Key Characteristics of a Mass Customizer (Boynton et al. 1993)

Change conditions	Constant and unforecastable changes in market demand, periodic and forecastable change in process technology
Strategy	Low-Cost process differentiation within new markets
Key organizational tool	Loosely coupled networks of modular, flexible processing units
Workflows	Customer / Product specific value chains
Employee roles	Network coordinators and on-demand processors
Control system	Hub and Web system; centralized network coordination, independent processing control
I/T alignment challenge	Integration of constantly changing network information processing/-communication requirements; interoperability, data communication and co-processing critical to network efficiency
Critical synergy	Reliance on continuous improvement form for increasing process flexibility within processing units

2.3 Characterization of PSS

Müller defines PSS as customer, life-cycle and sustainability-oriented socio-technical systems, solutions or offerings that integrate both products and services. The resulting business relationships integrate the customer as well as the vendor and aim at providing functionality to meet customer needs. As a success factor for PSS development and implementation, he points at the ability to adapt quickly and efficiently to changing customer requirements and to anticipate these changes already at an early stage of product development. Another one is the efficient capture and monitoring of customer needs over time (Müller 2013).

Meier focuses on industrial PSS in his work. Here, he also characterizes the joint development of product and service components as a key feature. He also identifies the adaptability of the solution to changing requirements in the product life-cycle as well as the possible exchange and reconfiguration of individual components of the PSS (Meier et al. 2010).

Morelli also sees the use of PSS mainly between companies and not in the offer for the consumer. For him, the PSS is the result of a so-called value co-production, which is operated out of a value-added network on the basis of a common development process between supplier and customer (Morelli 2006).

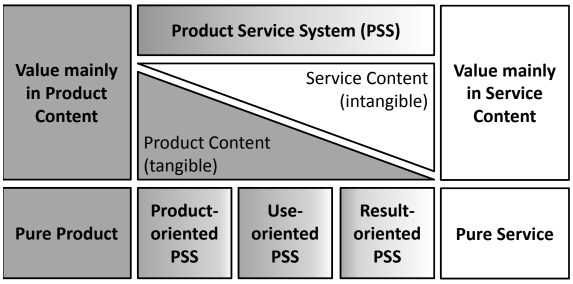

Fig. 2: Main Categories of PSS (Tukker 2004)

Tukker sets up a framework for the characterization of different PSS, in which he differentiates product-oriented, use-oriented and result-oriented PSS and the resulting business models (Fig. 2). As an example of result-oriented PSS he introduces pay-per-print models in which the customer is in principle not involved in the development of the printing systems but specifies the print result and other requirements. Furthermore, Tukker evaluates the impact on the market value of the offered solution for the customer, costs for the provider, capital expenditure and mutability for eight formulated PSS types (Tukker 2004).

In her description of the PSS concept, Mont emphasizes the benefit of PSS for manufacturing companies and postulates an extension of the product life-cycle. In addition to an intensification of the customer-interaction process during collection and monitoring of requirements, an additional value for the customer results from a possible upgrade and modernization of the product itself. Prerequisite is a suitable product architecture, which allows disassembly and disposal or repair and re-marketing of the product or its individual components (Mont 2002). With this, Mont takes up the thoughts of Wohlgemuth-Schoeller on the effects of modularization on the recycling of products. She argues that modular product architectures are highly beneficial for re-utilization, because they enable rapid diagnosis, repair and thus a quick replacement of a used product on the market. On module and component level the use as spare parts or the sale on module exchanges etc. is facilitated (Wohlgemuth-Schöller 1999).

2.4 Classification of PSS into the Product-Process-Change-Matrix

In relation to the product-process-change matrix presented in Sect. 2.1, PSS are classified in the following with respect to the change types according to the previous characterization.

With regard to the product or the offered service, PSS imply a change in customer needs over time. This must be taken into account when developing PSS, but the kind, extent and timing of these changes cannot be anticipated in advance. In the model of the product-process-change matrix, this corresponds to a dynamic product change.

Related to the company's internal processes for the synthesis, production and distribution of the customer-specific solution, these are largely stable. On the one hand, this is due to a rapid response to changed customer requirements. On the other hand, the life-cycle management of PSS also requires this stability with respect to the later disposal or recycling of PSS components as addressed by Mont.

In the product-process-change matrix, as presented in Sect. 2.1, there would be no difference between MC and PSS on the basis of the criteria of product and process change. Both represent a dynamic offering change with a stable process change. For a better differentiation the existing typology has to be extended by a third axis (Fig. 3).

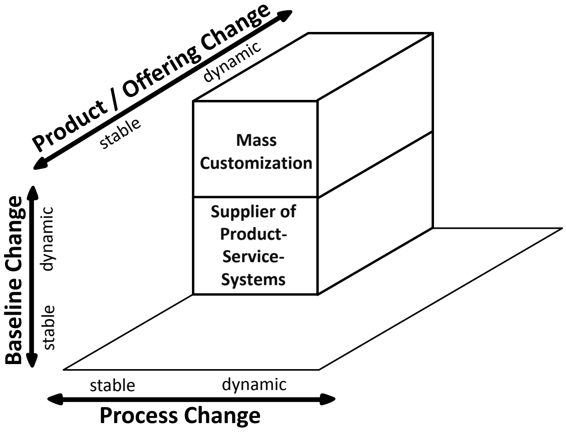

Fig. 3: Product-Process-Baseline-Change Matrix

The term *baseline* originates from configuration management and specifies a defined product variant, from which following states can be derived from and compared with. Using that, changes can be evaluated and documented more efficiently (Guess 2006).

A dynamic change in the baseline and at the same time a stable process and a dynamic product change rather leads to a replacement of a product already in use. Here, the mass customizer creates a new solution according to current customer requirements. On the contrary, a stable baseline change allows the supplier to react by adapting existing, possibly already delivered product and service components, as in the case of a PSS. Especially for use- and result-oriented PSS according to Tukker the baseline may not be changed ad hoc. Thinking e. g. of a car-sharing supplier, replacing the whole fleet at once in order to adapt to a new requirement setup is economically not meaningful since, like for tools, jigs and infrastructure in traditional product manufacturing, complexity costs behave highly remanent (Schaffer 2010).

At this point, the authors point out that this view does not mean that MC should be used for the consumer goods business, whereas PSS is more likely in the investment goods industry. First, such a simplification does not take into account the complex market relationships that arise with these types of businesses. Secondly, the delimitation between both may be hard to formulate in some cases. An example for this is the American grocery-shopping and delivery service *Peapod*. The business model that Peapod runs may be described as virtual supermarket. In order to distinguish them from other on-line grocery shops who more or less automate the trip to the retail store, Peapod lets the consumer customize their own virtual store so that they can shop in a way they want to. Users may arrange shopping lists, create favorites and organize for couponing and special offers. Moreover they can choose the way of delivery. On the other hand, due to a close interaction between customer and supplier, Peapod is able to track customer preferences and to forecast demands within certain borders (Pine et al. 1995). Although the organizational design of Peapod as described by Pine et al. is that one of a mass customizer, where the service components of the shopping experience are individualized, the whole system has some outlines of a result-oriented PSS.

Based on the conceptual similarity of the two business types, the hypothesis is made, that both the development processes and modeling tools used in mass customization can be applied on PSS. Therefore, the following section describes the state of the art with regard to the development and computer aided engineering environments for the single PSS constituents in order to investigate the PSS configuration afterwards.

3 Development and Configuration of PSS-Artifacts

The core of the PSS concept is the coequal development of product and service components. Therefore, the following section presents available development processes for an integrated development of PSS. Afterwards configuration and reconfiguration of PSS is discussed. Based on this, computer aided engineering environments, i. e. computer-aided design tools for PSS components, are presented.

3.1 Integrated Design Processes for PSS

Processes for the integrated development of goods and services within the meaning of a PSS have been discussed in the literature for about ten years. The majority of publications is restricted to partial aspects of the development processes or individual components, either product or service parts, of a PSS (Aurich et al. 2006, Spath and Demuß 2006, Yang et al. 2009, Vasantha et al. 2012). Integrated PSS development processes are only discussed in an isolated character.

Müller is developing a process-oriented approach from the point of view of systems engineering. The approach of the "layer-based PSS development" presented by him is based on the V-model XT and combines the different perspectives "PSS life-cycle", "PSS architecture" and "PSS development management". It reflects on the one hand the development of building blocks decomposed from the entire system as well as the system integration and validation out of these individual building blocks within the final PSS. The underlying framework is introduced as 150 % process, which has to be tailored and configured specifically to the needs of a specific PSS development task (Müller 2013).

In his work, Morelli provides various processes for the PSS development, which are mainly based on so-called "blueprints", i. e. on the workflows and flowcharts of various already successfully planned PSS (Morelli 2006). It is thus based on an established process for the development of physical products and software artifacts, the so-called templates. A template may be understood as a parametric, updatable, and reusable building block within a digital prototype (Cox 2000). With regard to physical products, e. g. geometry templates are further distinguished into rigid and variable geometry templates. The first represent carry-over-parts or library components that have additional process parameters available which cover knowledge about application, design interfaces or technical data in general. The latter is taken as predefined starting point for embodiment or detailed design that includes all necessary design rules and features (Hirz et al. 2013).

Steinbach presents a framework for the development of PSS, which is based on the distinction between structure-describing characteristics and behavioral properties. It is based on the idea of Weber's Property-Driven-Development (PDD) where the developer indirectly determines the properties of the product by defining the product characteristics based on his requirements (Weber 2005). With respect to the PSS's service components, the characteristics determine the potential and process dimensions, while the properties represent the result dimension of the PSS. Steinbach first concretizes this development process and then develops a software tool to support the designer (Steinbach 2005).

A review of further aspects for PSS development is presented by Cavalieri and Pezzotta (Cavalieri and Pezzotta 2012). They also provide an overview of individual methods and tools taken from the development of physical products. However, in this overview modular product and service architectures as well as product configuration are not mentioned.

3.2 Configuration of PSS

The configurability of PSS has already been discussed in literature in several places. Laurischkat focuses in her work on the configuration of the service components of a PSS. She assumes that based on five basic types of PSS service components, a so-called generation (which is equal to configuration) can be made based on the criteria of value proposition, life-cycle phase, reference and allocation, legal liability, case distinction, remote support, degree of automation and accountability. Through these criteria, service components can be connected to the functions of product components of a PSS e. g. through the use of configuration rules (if-then rules) or decision tables, thus opening up a solution space (Laurischkat 2012).

Mannweiler synthesizes a configuration process for industrial PSS, in which predefined PSS building blocks (mainly product components) are aggregated to a PSS based on customer requirements. The configuration is thereby measured at the degree of fulfillment, which shows how far the initially formulated requirements have been fulfilled by the specific configuration (Mannweiler 2014).

Aurich et al. examined the configurability of PSS in a more general way. They focus on the possible product and service architectures for a PSS (Aurich et al. 2006)(Aurich et al. 2009). The approach is largely based on the idea of modularization. For the synthesis of the customer-specific PSS variant, the authors apply the principle of the configuration and compatibility matrix (Puls 2003).

In this context, modularization of service components is examined by Lubarski and Pöppelbuß. In their framework, they go beyond the usual presentation of individual modularization projects for specific processes and the method applied for this purpose. The focus thus shifts to a fundamental discussion in which the phases of modularization (information capturing, decomposition, structuring, module creation, interface definition and testing) are related to the structuring level (logical, temporal or combined / complex structure) (Lubarski and Pöppelbuß 2016).

3.3 Computer Aided Engineering and Design of PSS Components

As PSS can be seen as aggregation of product and service constituents that fulfill individual customer needs, both components and the whole system have to be developed and designed. In this subsection, the computer-aided design and engineering for products, services and whole PSS are briefly introduced for traditional and knowledge-based modeling.

The role of the computer in modern product development is generally accepted. Today, complex products are modeled in a computer-aided engineering environment (Vajna et al. 2009). As computer-aided engineering environment, the authors of this paper understand a toolbox for the development of domain-specific artifacts, which includes the necessary tools for all synthesis and analysis activities as well as their information technology interfaces and data repositories. The according systems are used to determine the product design (mechanical CAD, MCAD) and to derive the necessary production data in the sense of technical drawings (Hirz et al. 2013). Another important aspect in terms of resource efficiency, functional integration, and cost-effectiveness of product components is their computer-aided simulation and optimization.

With respect to the computer-aided design of services (Service-CAD, SCAD) so far, only individual approaches have been documented. Sakao et al. developed the Service Explorer to provide a computer-aided service modeling tool based on a vendor-consumer system (Sakao et al. 2009). In this system, first the requirements and the state of a buyer are modeled. Afterwards the transformation rules into a desired state follow. This is realized by decomposed functional units of the service provider, similar to the feature-based modeling in geometry design. The principle behind this corresponds to the modeling of functional structures as they were used for physical products in the 1980s and 1990s by Roth (Roth 2000).

Hara et al. imply that modeling of customer benefit is not possible in CAD systems for physical products (Hara et al. 2006). However, this statement is not true per se due to the possibilities of parameterization and the implementation of knowledge in today’s CAD systems. The fulfillment of quantifiable requirements and the resulting usefulness can indeed be integrated within digital product models (Gembarski et al. 2015).

Knowledge-based modeling of physical components is today state of the art and a step beyond traditional parametric or feature-based modeling (Fig. 4). The latter means that certain functional elements like stiffening ribs or milled pockets are formulated as feature that instantiates geometry, parameters and behavior. These features are then implemented as design elements within the current CAD-model so that the user saves time. In contrast to that, knowledge-based modeling offers the possibility to automate aspects of the entire design process since the modeling system has

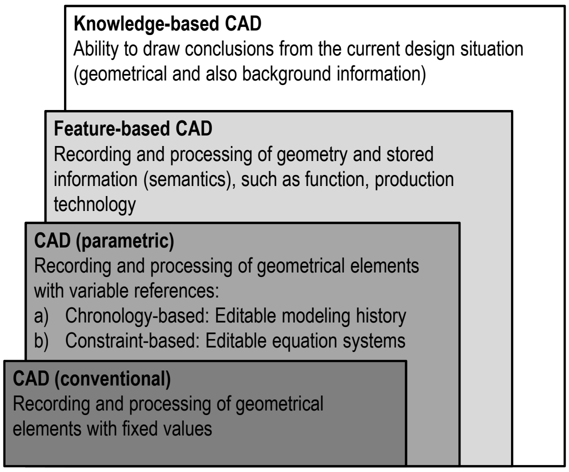

Fig. 4: Overview of the principles of 3D modeling (acc. to VDI 2209-03 2009)

the ability of reasoning and drawing conclusions (Gembarski et al. 2015).

Related to service modeling, a life-cycle-oriented approach for the knowledge-based assignment of service modules is provided by Yang et al. (Yang et al. 2009). According to their considerations, service modules can be linked to a product which is prepared for this purpose. The modules are triggered on the basis of data that is monitored during product use. As an example, they present monitoring a game console for acceleration and mechanical shock. In the event of shock, potentially damaged components of the console can be replaced quickly, without the need for an additional diagnosis step in customer service. Details on the design of the necessary knowledge base for the evaluation of events or reasoning mechanisms, as they are typical for knowledge-based systems (see also Sect. 4.2), are not given.

With the service design catalog, Akasaka et al. provide an extension for the Service Explorer (Akasaka et al. 2012). The catalog described there is developed as a support system for the synthesis of service parts of a PSS, which provides the developer with service modules for functions to be implemented. According to their own statement, the authors followed Roth's design catalogs that were developed in the 1980s and 1990s as knowledge base for design knowledge (Roth 2001). The typical structure of a design catalog consisting of index structure, main part and selection characteristics is not yet visible.

In her work, Kuntzky presents a knowledge-based development system for PSS, which is based on the technique of case-based reasoning (Kuntzky 2013). The basis for this is the modular design of the PSS components, as well as the formulation of requirements and knowledge about the aggregation of a particular PSS. Based on this data, a configuration of PSS is possible in the early phase of the development, if the same or similar PSS and its requirement profiles can be found in the case base for further processing. In contrast to other techniques of knowledge-based systems, knowledge does not have to be translated into a formal, explicit model (see also Sect. 4.2).

3.4 Intermediate Result

Individual integrated development processes for PSS can be identified in literature. However, it should be noted that these approaches remain very vague and conceptual in their application and validation to specific PSS development projects. Furthermore, they are partly discussed on very simple examples, which makes the transfer to the development of practice-oriented, complex PSS more difficult. Likewise, the approaches often neglect the important feature of the planning and design for configuration and reconfiguration of a PSS over the product life-cycle what has been identified as a central aspect in the development of PSS. At least, Müller gives the reference to an accompanying configuration management according to ISO 10007, like it is applied today, e. g. in the aircraft industry.

The configuration of PSS is generally regarded as feasible. Often, prerequisite are modular product and service architectures. Reconfiguration in the use phase of the product life-cycle based on changed requirements is, however, only explicitly included in individual approaches of service modeling. However, the documented configuration models show weaknesses in the application for complicated multi-variant systems, which has been reported in particular with respect to configuration and compatibility matrices in literature.

A joint modeling of product and service components in the sense of a common, parametric data model has not been documented until now and interfaces between MCAD and SCAD are currently not investigated. Individual software prototypes are presented for computer support of the configuration process, but the used configuration mechanisms and reasoning techniques are presented only superficially or represent more or less simple production rules. In this context, it should be noted that purely rule-based systems were excluded as a paradigm for the configuration of complex technical systems, because they are too inflexible in their design and expensive to maintain. A link to the knowledge technologies or to KBE is not drawn. An exception to this is the approach of case-based reasoning presented by Kuntzky. But, the knowledge about the context of problem-solving and solution is implicitly modeled and not by a parametric data model in the sense of a coupling between MCAD and SCAD.

The specific modeling of a design solution space, which represents product and service components of a PSS together with its dependencies and parameters, could not be researched from literature so far. Such design solution spaces are often represented by product configurators as used for MC. Thus, these are presented fundamentally and characterized as knowledge-based systems in the following section.

4 Solution Space modeling with Product Configurators

For the modeling of design solution spaces and for presentation of these solution spaces to the customer, product configuration systems are a suitable tool (Forza and Salvador 2006). Since configuration of products in MC is well established, the following description will be focused on product centered offerings. As stated in the last section, knowledge-based modeling and configuration of PSS is regarded feasible and will be addressed at the end of this section.

4.1 Application of Configuration Systems

Configuration is originally understood as development activity in which an artifact is formed by the aggregation of predefined building blocks. Those can only be connected via standardized interfaces and communicate in a predefined manner (Sabin and Weigel 1998). Nowadays, the meaning of configuration has to be extended because parametric design allows the definition of degrees of freedom like variable dimensions for a product component which are determined in the later configuration process. Another important aspect of parametric design is the ability to define relationships and constraints between parameters and product properties. So, a product variant can also be checked against technical or economic restrictions that are implemented in the configuration system (Gembarski and Lachmayer 2017). This allows the product configurator to be more than just a filter that is applied to an existing product portfolio to find exactly one or even no end product variant that matches to the given requirements. Configuration systems include a knowledge base in which the design knowledge is stored, indicating whether two options are mutually exclusive, whether the selection of a system component leads to adjustments to the current configuration, or whether the geometry parameters of a product are valid.

The aforementioned feature leads to the use of configurators as a sales support system. The main function of sales configurators is the distinct translation of customer requirements into a technical specification. Other functions include calculation, generation of quotation documents and the visualization of the end-product.

Today’s sales configurators also allow tracking all steps the user has taken in the configuration process. In detail, this includes sequence and duration of the single steps, as well as the termination and resumption of the configuration process in total. From this data, important information for distribution in relationship to trend scouting or preference analysis of different product variants can be obtained (Pine and Davis 1993).

Highly developed sales configurators, so-called choice navigators, even allow a bidirectional communication with the customer, so that a customer may be directed towards an, e. g., popular product variant. The basis for this are, for example, personal data collected from the customer beforehand, statistical data or data from social networks. This is intended to simplify the configuration process, since the customer is already confronted with a baseline or start configuration, which in general corresponds to his or her needs and is only adapted in small parts afterwards. On the other hand, a customer can also be influenced in the sense that "other customers who describe themselves as sportive have opted for this and that configuration" (Gembarski and Lachmayer 2015b).

In contrast to sales configurators, design configurators are mainly developed for internal use within a product development department. Such configurators are basically knowledge-based systems and are aimed at transforming a design problem into a configuration problem. For this purpose, all the necessary design knowledge is explicitly stored in the system, regardless of incorporating design rules, guidelines or manufacturing restrictions (Gembarski et al. 2015).

These expert systems do not replace the product developer, but they support in developing complex technical systems that could rather not be modeled without the computer-aided technologies.

4.2 Configurators as Knowledge-based Systems

Design configurators are generally classified as part of knowledge-based engineering systems (KBES). As such, they provide product descriptions based on predefined functions, components, constraints, relationships, and preference criteria. Chapman and Pinfold understand KBE as an evolutionary step in computer-aided product development, which link object-oriented programming, artificial intelligence, and computer-aided design to generate automated solutions for variant design (Chapman and Pinfold 2001).

As a sub-group of knowledge-based systems, KBES consist of the following components (Milton 2008):

1. Knowledge Base: Container for all types of declarative and domain-specific information, structures and rules. In the context of mechanical design, these can be dimensions of a standard part or manufacturing restrictions.
2. Inference Engine: Separated type of knowledge that controls the exploration of the design solution space by the KBE system. It describes both individual inferences, i. e. the application of the knowledge base for calculation or evaluation, as well as task knowledge, i. e. the linking of inferences, user inputs and AI methods such as constraint propagation to complex planning and design tasks, e. g. for the design of elevators or cement factories.
3. Blackboard: Working memory for the instantiation of case-specific parts of the knowledge base and intermediate results.
4. User Interface: This allows the interaction between user and system.
5. Editor: This allows the interaction between the knowledge engineer and the system to alter the knowledge base and the inference engine.

Especially for KBES an integration of a computer-aided engineering environment is necessary which is commonly realized by implementation of CAD systems (La Rocca 2012).

With respect to the inference mechanism, three different approaches are fundamentally differentiated:

- Rule-based: The representation of knowledge is based on production rules, which are formulated as if-then-else statements. These rules do not have to be related to each other (but can be: rules can be used to initiate subordinate rules or to eliminate other rules from the working memory). Many authors report that purely rule-based systems are only suitable for use with local and narrowly defined problems. This is due to the fact that with an increasing number of artifacts and rules, these systems suffer bad maintainability (McDermott 1982).

- Model-based: The limitation of the design solution space is based on a product model consisting of the system components and their relationships. The relationships may be e. g. physically or logically (constraint-based) or on the basis of resource allocation and resource consumption (Heinrich and Jüngst 1991).
- Case-based: Here, no explicit configuration rules or models are needed. The reasoning is made on the basis of previously recorded solutions (cases). Depending on the maturity of the inference mechanism, the system can either only find solutions that exactly match to a given requirement profile, or make a selection of several cases representing the best-fit. Highly developed systems are able to alter or combine existing cases in order to derive new solutions (Gembarski and Lachmayer 2015b).

4.3 Configuration of PSS

The existing literature reports occasionally the implementation of product configurators for PSS development. As described in Sect. 3.4, these follow either rule-based (e. g. Laurischkat and in a broader sense Mannweiler) or case-based modeling paradigms (e. g. Kuntzky). Model-based approaches are currently not available.

Nevertheless, viable concepts for the configuration of both physical and non-physical artifacts on the basis of requirements and task knowledge can be identified from the literature of the 1990s. An example of this is XRAY, an expert system that has been designed for the prototypical development of X-ray analysis systems (Cunis et al. 1991). Mentioning such an old technology seems initially anachronistic, but XRAY is a well-documented expert system regarding its functionality, architecture and implementation.

XRAY was developed in PLAKON, an expert system core that, like software development environments today, provides all the necessary functions and classes for the creation of planning and configuration systems, including inference and conflict resolution mechanisms.

XRAY had to fulfill the following requirements:

- Interactive definition of the test task under consideration of the specimen geometry, defects to be detected, test time and resulting costs.
- Automatic selection and configuration of the hardware components for the X-ray inspection system as well as indication of alternatives and their effects on the test quality.
- Automatic generation of the test plan and an ideal test sequence in which the test should be carried out with minimized redundancies.
- Automatic configuration of the software for image recognition for efficient identification of possible errors and defects.
- Interactive simulation and test of the analysis software using sample images.

The developed system was able to meet these requirements in principle. Special attention must be paid to the common configuration of hardware and software components, which has been essential for the efficient performance of the test tasks. Model-based reasoning approaches were used predominantly as inference mechanisms. However, the project did not focus on connecting XRAY to a MCAD system, for example to generate production drawings.

On the product side, the bill-of-materials of the inspection system was an output. Regarding the service of the supplier, the setup, configuration and verification of the software was automated as well as consulting activities because the expert system tested alternative hardware- / software configurations and evaluated them with respect to time, cost and result quality. Additionally, calculating a test setup and sequence can also be regarded as service activity that has to be maintained either by an expert of the supplier or by the application engineer of the customer. This activity was widely automated as well.

5 Concluding Remarks

In this article, the business types of mass customization and supplier of PSS were compared and the basic applicability of product configuration at PSS was discussed. It was shown that single approaches for rule-based and case-based configuration already exist, but the model-based configuration has not been implemented so far.

The reason for this is the lack of a common data model for all artifacts of a PSS, regardless of whether it is hardware, software or service components, as well as for the relationships among each other. It would be desirable to build a parametric model so that adaptation and variant design of PSS can be performed in a similar way like it is already the state of the art in CAD systems available for physical products. Such a data model would additionally enable computer-aided product optimization. In this context, it is necessary to examine the extent to which Steinbach's approach to the transfer from the property-driven-development of PSS is a basis for this. A possible use case in a product-oriented PSS may be, for instance, the calculation of maintenance intervals on the basis of data about the used standard parts like bearings and the applied forces in the product use. A next step could be to automate the generation of spare part and tool lists as well as instructions and manuals for the maintenance procedure itself. On the other hand, by analyzing the neighborhood relationships of components in a CAD assembly, times for dis-assembly and re-assembly may be calculated automatically.

Another point of interest is the implementation of sales configurators, either as a support system for the representative in a sales department or as front-end system for customers. This is an important fact since the ability of configuring a PSS can be understood as service itself, which delivers a user experience. Especially in business-to-consumer markets this can even be extended by the methods of event marketing where the configurator is used, for example, as platform for a design contest.

References

Akasaka F., Nemoto Y., Kimita K., Shimomura Y. (2012) Development of a knowledge-based design support system for Product-Service Systems. In: Computers in Industry 63(4), pp. 309–318

Aurich J., Fuchs C., Wagenknecht C. (2006) Life cycle oriented design of technical Product-Service Systems. In: Journal of Cleaner Production 14(17), pp. 1480–1494

Boynton A., Victor B., Pine II B. (1993) New competitive strategies: Challenges to organizations and information technology. In: IBM Systems Journal 32(1), pp. 40–64

Cavalieri S., Pezzotta G. (2012) Product-Service Systems Engineering: State of the art and research challenges. In: Computers in industry 63(4), pp. 278–288

Chapman C., Pinfold M. (2001) The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure. In: Advances in Engineering Software 32(12), pp. 903–912

Cox J. (2000) "Product Templates" A Parametric Approach to Mass Customization. In: Brunet P., Hoffmann C., Roller D. (eds.) CAD Tools and Algorithms for Product Design. Springer-Verlag, Berlin, pp. 3–15

Cunis R., Günter A., Strecker H. (1991) Das PLAKON-Buch. Springer-Verlag, Berlin

Da Silveira G., Borenstein D., Fogliatto F. (2001) Mass customization: Literature review and research directions. In: International journal of production economics 72(1), pp. 1–13

Dabic M. (2006) Kosten und Nutzen der Individualisierung bei der Produkt-und Markenwahl für den Konsumenten: Eine empirische Studie am Beispiel des Automobilmarktes. PhD thesis, Wirtschaftsuniversität Wien

Davis S. (1987) Future perfect. Addison-Wesley Publishing co., Boston

Fogliatto F., da Silveira G., Borenstein D. (2012) The mass customization decade: An updated review of the literature. In: International Journal of Production Economics 138(1), pp. 14–25

Forza C., Salvador F. (2006) Product information management for mass customization: connecting customer, front-office and back-office for fast and efficient customization. Palgrave Macmillan, Houndmills, Basingstoke, Hampshire, United Kingdom

Gembarski P., Lachmayer R. (2015a) A Business Typological Framework for the Management of Product Complexity. In: Bellemare J., et al. (eds.) Managing Complexity – Proceedings of the 8th World Conference on Mass Customization, Personalization, and Co-Creation (MCPC 2015). Springer International Publishing, Switzerland, Montreal, Canada, pp. 235–247

Gembarski P., Lachmayer R. (2015b) Degrees of Customization and Sales Support Systems – Enablers to Sustainability in Mass Customization. In: Weber C., et al. (eds.) DS 80-1 Proceedings of the 20th International Conference on Engineering Design (ICED 15), vol 1: Design for Life. Design Society, Milan, Italy, pp. 191–198

Gembarski P., Lachmayer R. (2016) Mass Customization und Product-Service-Systems: Vergleich der Unternehmenstypen und der Entwicklungsumgebungen. In: Thomas O., Nüttgens M., Fellmann M. (eds.) Dienstleistungsmodellierung (DLM 2016). Springer, Berlin, Heidelberg, pp. 214–232

Gembarski P., Lachmayer R. (2017) Designing Customer Co-Creation: Business Models and Co-Design Activities. In: International Journal of Industrial Engineering and Management (IJIEM) 8(3), pp. 121–130

Gembarski P., Li H., Lachmayer R. (2015) KBE-Modeling Techniques in Standard CAD-Systems: Case Study – Autodesk Inventor Professional. In: Bellemare J., et al. (eds.) Managing Complexity – Proceedings of the 8th World Conference on Mass Customization, Personalization, and Co-Creation (MCPC 2015). Springer International Publishing, Switzerland, Montreal, Canada, pp. 215–233

Gräßle M., Thomas O., Fellmann M., Krumeich J. (2010) Vorgehensmodelle des Product-Service Systems Engineering – Überblick, Klassifikation und Vergleich. In: Böhmann T. (ed.) Proceedings der Teilkonferenz Integration von Produkt & Dienstleistung – Hybride Wertschöpfung im Rahmen der Multikonferenz Wirtschaftsinformatik. Books on Demand, Göttingen, Germany, pp. 31–42

Guess V. (2006) CMII for Business Process Infrastructure. Hollis Publishing Company

Hara T., Arai T., Shimomura Y. (2006) A Concept of Service Engineering: A Modelling Method and A Tool for Service Design. In: International Conference on Service Systems and Service Management. IEEE, Troyes, France, pp. 13–18

Heinrich M., Jüngst E. (1991) A resource-based paradigm for the configuring of technical systems from modular components. In: Proceedings of the 7th IEEE Conference on Artificial Intelligence Applications Vol. 1. IEEE, pp. 257–264

Herlyn W. (2012) PPS im Automobilbau – Produktionsprogrammplanung und Steuerung von Fahrzeugen und Aggregaten. Carl Hanser Verlag, München

Hirz M., Dietrich W., Gfrerrer A., Lang J. (2013) Integrated computer-aided design in automotive development. Springer-Verlag, Berlin

Kuntzky K. (2013) Systematische Entwicklung von Produkt-Service Systemen. Vulkan-Verlag, Essen

La Rocca G. (2012) Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design. In: Advanced engineering informatics 26(2), pp. 159–179

Lachmayer R., Gembarski P., Gottwald P., Lippert R. (2015) The potential of product customization using technologies of additive manufacturing. In: Bellemare J., et al. (eds.) Managing Complexity – Proceedings of the 8th World Conference on Mass Customization, Personalization, and Co-Creation (MCPC 2015). Springer International Publishing, Switzerland, Montreal, Canada, pp. 71–81

Laurischkat K. (2012) Product-Service Systems: IT-gestützte Generierung und Modellierung von PSS-Dienstleistungsanteilen. Shaker-Verlag, Aachen

Lubarski A., Pöppelbuß J. (2016) Methods for service modularization – A systematization framework. In: Proceedings of the Pacific Asia conference on information systems (PACIS 2016), Chiayi, Taiwan

Mannweiler C. (2014) Konfiguration investiver Produkt-Service-Systeme. PhD thesis, Lehrstuhl für Fertigungstechnik und Betriebsorganisation, Technische Universität Kaiserlautern

McDermott J. (1982) R1: A rule-based configurer of computer systems. In: Artificial intelligence 19(1), pp. 39–88

Meier H., Roy R., Seliger G. (2010) Industrial product-service systems – IPS 2. In: CIRP Annals-Manufacturing Technology 59(2), pp. 607–627

Miles R., Snow C. (1986) Unternehmensstrategien. McGraw-Hill, New York

Milton N. (2008) Knowledge technologies, 3rd ed. Polimetrica, Monza

Mont O. (2002) Clarifying the concept of product-service system. In: Journal of cleaner production 10(3), pp. 237–245

Morelli N. (2006) Developing new product service systems (PSS): methodologies and operational tools. In: Journal of Cleaner Production 14(17), pp. 1495–1501

Müller P. (2013) Integrated engineering of products and services: Layer-based development methodology for product-service systems. PhD thesis, Fraunhofer IPK, Technische Universität Berlin

Piller F. (2004) Mass customization: reflections on the state of the concept. In: International journal of flexible manufacturing systems 16(4), pp. 313–334

Pine B., Davis S. (1993) Mass customization: the new frontier in business competition. Harvard Business Press

Pine B., Peppers D., Rogers M. (1995) Do you want to keep your customers forever? Harvard Business Press

Pine B., Victor B., Boynton A. (1993) Making mass customization work. In: Harvard Business Review 71(5), pp. 108–117

Puls C. (2003) Die Konfigurations- & Verträglichkeitsmatrix als Beitrag zum Management von Konfigurationswissen in KMU. -Verlag, Düsseldorf

Reichwald R., Piller F. (2009) Interaktive Wertschöpfung. Gabler, Wiesbaden

Renner I. (2007) Methodische Unterstützung funktions-orientierter Baukastenentwicklung am Beispiel Automobil. PhD thesis, Technische Universität München

Roth K. (2000) Konstruieren mit Konstruktionskatalogen: Band 1: Konstruktionslehre. Springer-Verlag, Berlin

Roth K. (2001) Konstruieren mit Konstruktionskatalogen: Band 2: Konstruktionskataloge. Springer-Verlag, Berlin

Sabin D., Weigel R. (1998) Product configuration frameworks-a survey. In: IEEE intelligent systems 13(4), pp. 42–49

Sakao T., Shimomura Y., Sundin E., Comstock M. (2009) Modeling design objects in CAD system for service/product engineering. In: Computer-Aided Design 41(3), pp. 197–213

Schaffer J. (2010) Entwicklung und Optimierung eines treiberbasierten Modells zur Bewertung varianteninduzierter Komplexitätskosten in industriellen Produktionsprozessen. Sierke Verlag, Göttingen

Schallmo D. (2013) Geschäftsmodelle erfolgreich entwickeln und implementieren. Springer-Verlag, Berlin

Spath D., Demuß L. (2006) Entwicklung hybrider Produkte—Gestaltung materieller und immaterieller Leistungsbündel. In: Service Engineering. Springer-Verlag, Berlin, pp. 463–502

Steinbach M. (2005) Systematische Gestaltung von Produkt-Service-Systemen – Integrierte Entwicklung von Produkt-Service-Systemen auf Basis der Lehre von Merkmalen und Eigenschaften. PhD thesis, Lehrstuhl für Konstruktionstechnik/-CAD, Universität des Saarlandes

Tukker A. (2004) Eight types of product-service system: eight ways to sustainability? Experiences from SusProNet. In: Business strategy and the environment 13(4), pp. 246–260

Vajna S., Weber C., Bley H., Zeman K., Hehenberger P. (2009) CAx für Ingenieure – Eine praxisbezogene Einführung. Springer-Verlag, Berlin

Vasantha G., Roy R., Lelah A., Brissaud D. (2012) A review of product-service systems design methodologies. In: Journal of Engineering Design 23(9), pp. 635–659

VDI 2209-03 (2009) 3-D-Produktmodellierung – Technische und organisatorische Voraussetzungen – Vefahren, Werkzeuge und Anwendungen – Wirtschaftlicher Einsatz in der Praxis. Verein Deutscher Ingenieure, Beuth, Berlin

Vogel-Heuser B., Lindemann U., Reinhart G. (2014) Innovationsprozesse zyklenorientiert managen: Verzahnte Entwicklung von Produkt-Service Systemen. Springer-Verlag, Berlin

Weber C. (2005) CPM/PDD – An extended theoretical approach to modelling products and product development processes. In: Bley H., et al. (eds.) Proceedings of the 2nd German-Israeli Symposium on Advances in Methods and Systems for Development of Products and Processes. Fraunhofer-IRB-Verlag, Stuttgart, Germany, pp. 159–179

Wohlgemuth-Schöller E. (1999) Modulare Produktsysteme. Peter Lang Verlag, Frankfurt am Main

Yang X., Moore P., Pu J.-S., Wong C.-B. (2009) A practical methodology for realizing product service systems for consumer products. In: Computers & Industrial Engineering 56(1), pp. 224–235