Design for customization

Elgammal, Amal; Papazoglou, Mike; Krämer, Bernd; Constantinescu, Carmen

Published in:
Procedia CIRP

DOI:
10.1016/j.procir.2017.03.132

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 64 (2017) 345 – 350

www.elsevier.com/locate/procedia

The 9th CIRP IPSS Conference: Circular Perspectives on Product/Service-Systems

Design for Customization: A New Paradigm for Product-Service System Development

Amal Elgammal[a,*], Mike Papazoglou[b], Bernd Krämer[a], Carmen Constantinescu[c]

[a]Scientific Academy for Service Technology e.V. (ServTech), Bürgerstr. 54a, Hagen D-58097, Germany
[b]European Research Institute in Service Science (ERISS), Tilburg University, Tilburg 5000 LE, Netherlands
[c]Fraunhofer Institute for Industrial Engineering, FhG-IAO, Stuttgart, Nobelstraße 12,Germany

* Corresponding author (Faculty of Computers & Information, Cairo University). Tel.: +201094488895. *E-mail address:* a.elgammal@fci-cu.edu.eg

Abstract

In the traditional software development cycle, requirements gathering is considered the most critical phase. Getting the requirements right first time has become a dogma in software engineering because the correction of erroneous or incomplete requirements in later software development phases becomes overly expensive. For product-service systems (PSS), this dogma and standard requirements engineering (RE) approaches are not appropriate because classical RE is considered concluded, once a product service is delivered. For PSS it is impossible to foresee all future context conditions and customization needs customers may come up with after product deployment. In addition, the services supporting a complex hardware-software product depend on the individual product configuration a customer requires. For example, when a standard laser machine is equipped with one or more special sensors, new services may be needed that depend on sensor data from these new sources combined with other data generated by the standard machine configuration. Thus, we claim that RE needs to be extended to the deployment phase of a product and an agile approach is required to cope with emerging hardware and software requirements as a PSS is marketed. In this paper, a novel view-based model-driven engineering approach is proposed that enables collaborative product-service design and customization and copes with evolving, incomplete and unforeseen requirements. A prototype has been implemented as a Proof-of-Concept (PoC) and is currently validated on four industrial pilot cases as part of the H2020 project ICP4Life.

Peer-review under responsibility of the scientific committee of the 9th CIRP IPSS Conference: Circular Perspectives on Product/Service-Systems.

Keywords: Mass Customization; model-driven Engineering; Domain Specific Language (DSL);View-based Modelling; Agile Product Service (PS) development.

1. Introduction

Mass production mainly involves the production of large amounts of standardized products, where customers have no longer the ability to individualize or influence the composition of the end product [1-3] . Product customisation recently became an emerging business need as companies began to recognise the great importance and benefits of delivering individualized customized products, while retaining the advantages of the mass production method. By having a greater focus on customers, companies can use this strategic opportunity to gain a competitive advantage and achieve a noticeable economic value [2, 4]. In their desire to become customer-driven, many companies have resorted to inventing new programs and procedures to meet every customer's request. But as customers and their needs grow increasingly diverse, such an approach adds unnecessary cost and complexity to operations. As a result, companies have embraced mass customization in an attempt to avoid those pitfalls [5].

Product design seen from a perspective that combines products with its corresponding services is often termed Product/Service Systems (PSS) [6]. In PSS a product corresponding service is regarded as conventional service activity (e.g., maintenance, repair) that is coupled with a physical product to meet customer requirements.

Currently, PSS are facing severe drawbacks. The most notable drawbacks are that PSS are prominently conceptual

Peer-review under responsibility of the scientific committee of the 9th CIRP IPSS Conference: Circular Perspectives on Product/Service-Systems.
doi:10.1016/j.procir.2017.03.132

taking a marketing or business perspective and lacking a concrete IT implementation. PSS do not accommodate evolving user preferences or product differentiation features. Current PSS are unable to cope with multiple stakeholder views and automatically tailor product design to a customer's needs in real-time. They also neglect the collaborative aspects of the product design and development process. More importantly, PSS cannot provide a holistic view of products associating product structure with product quality. They do not support the composition/substitution of product-parts into the design of a coherent final product, and do not support analysis of product-related data gathered along product lifecycles – which remains totally unexplored - to improve data-driven decision making.

Mass customization (MC) refers to the combination of mass production and the capability to offer customized products. For any successful implementation of mass customization, a *product customizer* is a core software component, where a *model* describing the desired product must be defined and implemented employing the product customizer functionality. Product models must be developed collaboratively to describe an intended physical artefact (product-service) that is usually incomplete in the form of partial models that can be completed by the interaction with user groups and different stakeholders.

This paper proposes a novel formal view-based model-driven engineering approach for collaborative product-service (PS) customization filling current gaps and pitfalls. The proposed approach accommodates various complimentary stakeholders' perspectives by fostering view-based modelling of the various aspects of the product, e.g., structural, operational, quality aspects, etc. The PS customization approach is founded on *formalism-based knowledge-intensive structures* called *production blueprints*, which capture rich product-service and production-related knowledge. This enables automated reasoning and inference to validate the consistency of the customizer PS, verify customer constraints and preferences and generate a preliminary production plan. In the heart of the customization approach is the Product-oriented Configuration Language (PoCL), a graphical Domain-Specific Language (DSL) aims at easing the collaborative product design task using the same jargon familiar to customers and other stakeholders, in an abstract and intuitive manner. The bi-directional mapping between PoCL models and production blueprints is defined and is preliminary validated on two real-life industrial pilot cases.

The rest of this paper is organized as follows: related work is summarized in Section 2 and appraised against the work proposed in this paper. Section 3 presents the view-based model-driven engineering approach. Section 4 introduces PoCL and its implementation. The added-value of production blueprints is discussed in Section 0. Conclusions and future work are sketched in Section 6.

2. Related Work

Mass customization is a synthesis between mass production and customized goods and services [4]. Pine [4] further distinguishes the following four types of customization: (i) Collaborative customization: firms interact with customers to determine the precise product offering that meets the customer's needs; (ii) Adaptive customization: firms produce a standardized product that can be altered and customized by the end users within given adaptation limits; (iii) Transparent customization: firms provide customers with unique products without telling them that the products are customized; and (iv) Cosmetic customization: firms produce a unique standardized product and market it to customers in different ways.

According to Joergensen et al. [7], the successful implementation of a customization strategy must include important factors such as: understanding the market needs, creating a modular product platform, postponement of variety until actual demand arises, and establishment of flexible production processes. Collaborative and adaptive customization are especially important for our objective as well as the factors described in [9] as they address the alignment between the market demands, the design of a product and the effective setup of production.

A product architecture describes the decomposition of a product into product modules and the arrangement and explicit interfaces between these modules [8]. Product architectures, also called product platforms, enable the development of product families and generations of products, using shared assets to enable cost-effectiveness. These shared assets also provide many operational benefits, for example, in parts sourcing, manufacturing, and quality control. A product family comprises a set of variables, features or components that remain constant in a product platform and from product to product. Platform-based product family design has been recognized as an efficient and effective means to realize sufficient product variety to satisfy a range of customer demands in support for mass customization [9].

To keep the cost of products close to that of mass production, customization is assured via modularity and commonality in product design. The ability to replace one module with another without changing the interface on either side is key to the effective creation of a product variety. In the design phase, commonality primarily acts to reduce the types of efforts required to produce a product variant [10]. Tian et al. [11] argue that in the case of mass customization a parameterised method may be used to plan the processes of the products and parts since great similarities exist between the products for mass customization and their components. In the parameterised process planning, process is automatically generated based on the values of the corresponding variables.

Research related to product customization has also focused on knowledge intensive models for representing product family platforms [12-14]. Zha and Sriram [12] present a module-based integrated design scheme to support product family architecture modelling, product platform establishment, product family generation, and product variant assessment. The information and knowledge-modelling system in this publication can be used for platform product design knowledge capture, representation and management and offer on-line support for designers in the design process. Several research works have concentrated on the concept of customer-centric enterprise to focus all company operations on serving customers and deliver unique value by considering customers as individuals [15]. At the operational level, mass customization and personalization have emerged as leading ideas in the last decade to reach the objective of empowering customers [16]. Giovannini et al. [17] present an innovative approach to rationalise the product variety by linking each product variant to the customer's profile who needs it. Their aim is to optimise the product variety avoiding excesses (variants not related to a customer), lacks (customers not related to a variant) or redundancies (two or more variants proposed to a customer). A knowledge-based system was designed to capture customer needs, functional constraints and design parameters.

While there is increasing interest in PSS, limited research has been done in the design and development aspects of PSS [18]. Authors is [19] identifies characteristics of various strategies that may be applied in PSS development. State-of-the-art in product-service systems has been presented in [20], where as opposed to the work proposed in this paper, no concrete formal-based model-driven engineering approach is proposed to concretely recognize this promising approach.

Mass customization implementations in the literature mainly focused on capturing direct information of the customization process representing the solution space. However, information that is not directly used to perform the customization, such as data regarding customers, suppliers and product quality information, is as important as direct product information. Therefore, the product service customization approach presented in this article is founded on formalised knowledge-intensive structures called *production blueprints*, which capture rich and standardized product-service and production-related knowledge. The notion of *production blueprints* is built upon the novel notion of *manufacturing blueprints* [21], which rely on model-based design techniques to manage and inter-link product data, information, product portfolios and product families, manufacturing assets (personnel, plant machinery and facilities, production line equipment), production processing requirements and workflows. Manufacturing blueprints help meet the requirements (functional, performance, quality, cost, physical factors, interoperability, time, etc.) of an entire manufacturing network. This information can be collated and put within a broader operational context, providing the basis for production actionable “intelligence” and a move toward more fact-based manufacturing decisions.

However, information modelling concepts and mechanisms may eventually become intractable and are generally difficult to use and understand by customers and product engineers [21, 22]. This represents a main obstacle of the successful implementation and realization of a product configurator based on a solid formal foundation, which can ensure the sound composition of the desired customized product and the satisfaction of customers' constraints and preferences. To surmount this *usability* issue, the product customization approach presented in this article utilizes model-based techniques to facilitate collaborative product customization and design. A model-based Product-oriented Configuration Language (PoCL) bridges this usability gap. PoCL is a graphical DSL to ease the collaborative product design task. An automated bi-directional mapping from PoCL models to formal blueprints is defined and implemented.

3. Proposed Approach

This section presents a conceptual view of the proposed view-based model-driven engineering approach for collaborative product customization. In model-driven engineering [23], two levels of models usually co-exist:

- *Platform Independent Model (PIM)*: is a model of a software system that is independent of the specific technology platform used to implement it. Fig. 1 presents the blueprint model-driven engineering approach. The PIM on the left hand-side of the figure is given in the form of PoCL to be further elaborated in the following section.
- *Platform Specific Model (PSM)*: is a model of software that links to a specific technology platform. Platform-specific models are indispensable for the actual implementation of a system. Following the blueprinting approach shown in Fig. 1, the PSM represents the production blueprints (right-hand side of Fig. 1), which capture a diversity of product and production-related knowledge implemented as an Ontology Web Language (OWL) ontology. Blueprints enable automated reasoning and verification to ensure the consistency of the customized product request and the satisfaction of a customer's requirements and constraints as OWL is formally founded on Description Logic.

The basic idea of model-driven engineering is a PIM can be systematically transformed into a platform-specific model (described below). Model-driven engineering, in general, has two main advantages: (i) It facilitates the work of targeted stakeholders by adding a layer of abstraction, which conforms to the jargons (concepts/terminologies) used in their daily work practices. The stakeholders do not need to know the low-level details of the platform specific language used. (ii) It needs only one PIM, which can be mapped into multiple PSMs that different end-users are agnostic of. The product configuration approach proposed in this paper and its realization in a product configurator (see Section 4) has the main objective of emphasizing ‘‘customer centric’’ aspects, which are commonly required by Product Service Systems (PSS) [6].

The bi-directional automated transformation between PIM and PSM is performed by a ‘transformation engine’ component (represented by the two arrows connecting the PIM and PSM realms shown in Fig. 1). From PIM to PSM (‘Transformation & Consistency checking’ arrow), automated transformation of PoCL models into the blueprinting formal representation are conducted. This enables the automated checking of the consistency of the customized product, which is then stored in a ‘Production Blueprint Repository’ to enable product variants reusability and further customization. From PSM to PIM (Reuse and Configuration arrow), the ‘Production Blueprint Repository’ can be queried and browsed by customers supported by a ‘Product Library’. This capacitates the re-usability of configured product variants that may be further configured to meet a new product instance specific customization needs. The proposed approach allows configuring customer orders by making use of PoCL. PoCL (PIM) and production blueprints (PSM) are briefly discussed next in Sections 4 and 5, respectively.

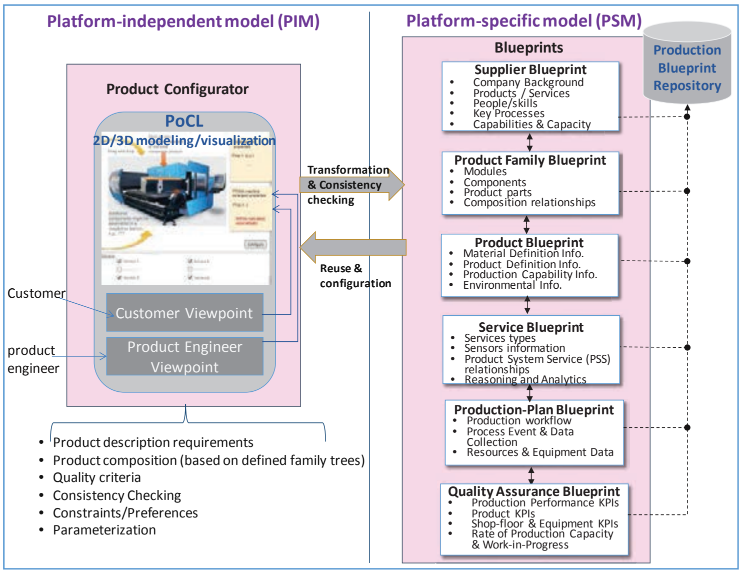

Fig. 1 The view-based blueprinting model-driven engineering approach for collaborative PS customization

4. Product-oriented Configuration Language

PoCL is a model-based graphical intuitive Domain-Specific Language (DSL) that helps resolve production design problems using an interactive graphical software framework. This framework allows a variety of stakeholders, such as customers, product engineers and service designers, to create conceptual drawings for final assessment and approval of a virtual product. It supports different types of complex innovative shapes and helps streamlining collaboration between stakeholders, while checking the *consistency* and *satisfaction of customer's requirements inline* with the customization process.

PoCL interacts with users depending on their level of expertise ranging from customers with design engineering knowledge, which are closely involved in the co-design of products, to fairly lay-users. The language employs easily combinable graphical representations of product shapes and combines product quality characteristics with product design artefacts. Different stakeholders are supported by different interfaces with varying abstraction levels that accommodate with their perspectives. Fig. 2 presents the meta-model of PoCL, represented as a UML class diagram, by focusing on the customer view. To enforce standard product and process descriptions and ensure wide applicability, the meta-model has been iteratively developed and validated based on common manufacturing and supply-chain standards. This includes: ISA-95 (www.isa-95.com), the Supply Chain Operations Reference model (www.supply- chain.org). Also, the ISO "Standard for the Exchange of Product" and STEP (http://www.steptools.com), providing standard descriptions of product data, and supports benchmarking against standardized performance measurements. Furthermore, the meta-model is being continuously validated and refined by interacting with industrial partners of the Horizon 2020 project ICP4Life (http://www.icp4life.eu). They include PRIMA, Laserlam, Bazigos and Tauron, which represent diverse industrial areas to ensure the approach's generality.

The 'Product' class in Fig. 2 represents the heart of the meta-model and plays a pivotal role. 'Product' is modelled as an abstract class (a class that cannot be instantiated directly). Class 'Product' has two main sub-classes: 'Base Product' and 'Non-configurable' product. Base Product' represents a configurable product and represents a starting point for configuration, and can also be requested as it is. 'Non-configurable' product is a product that cannot be configured. A customer may also select a base product and configure it to satisfy her specific needs. This case is represented in Fig. 2 by the 'Requested Product' class (a sub-class of 'Base Product'). 'Requested Product' represents a virtual product that is not yet manufactured. Once manufactured, the requested product becomes a 'Configured Product'. A customer may also start a configuration from a previously configured product (class 'Configured Product', which is inherited by class 'Requested Product'.

A 'Product' must belong to a 'Product Family'. Product family modelling is a commonly adopted approach for the design of customised products [2]. Product family is described as a single model and can be viewed as the setup of end products, which can be formed by combining a predefined set of modules or product parts. A product family model typically describes which modules are part of the product family model and how they can be combined with each other [24, 25]. Then the customer can start customizing two views of the product service: (i) *Visual parameters* (represented by the "Machine Visual Parameters" palette), which represents parameters related to the visual appearance of the PS. 'Product Part' can be either 'Inhouse Part' or 'Outsourced Part' from 'Inhouse Supplier' or 'Outsourced Supplier', respectively. To capture the concept of Product Service Systems (PSS), 'Service' class is linked to 'Product', with a M:N association.

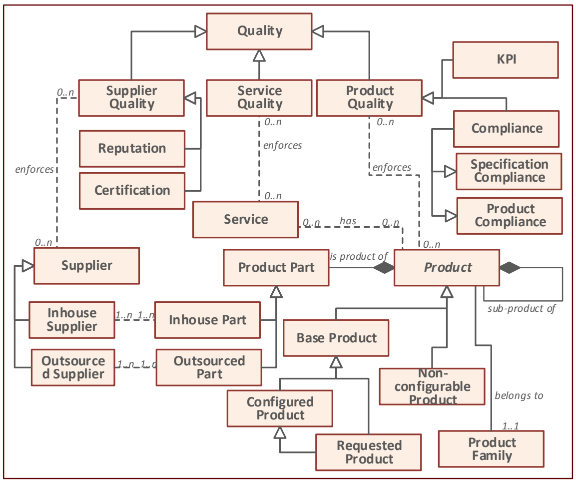

Fig. 2 PoCL meta-model represented as a UML class diagram

As quality is a vital aspect in PSS that must be supported, it is represented in the meta-model of Fig. 2 as class 'Quality'. Obviously, there are multiple dimensions of PS quality, which include: Service Quality, Product Quality (can be Key Performance Indicator (KPI) or Compliance) and Supplier Quality (can be Reputation or Certification).

Implementation: Fig. 3 shows a screenshot of the ICP4Life customizer implemented as a web-based app. PoCL is implemented as a graphical DSL as illustrated above. In particular, the screenshot supports the PRIMA/Laserlam pilot case. Usability is a major concern in the development of all components and GUIs. As shown in Fig. 3, the "PRIMA Machines" palette (left hand-side of the screenshot) dynamically shows the available base PRIMA machines. PRIMA's customer (Laserlam in our case) can select a base machine from the "PRIMA Machines" palette.

Suppose that Laserlam selects 'Platino' machine which comes with few sensors and supporting services. Now Laserlam needs to add a special sensor to a given product family and requires a signal, when both the vibration data surmount a certain wave length or amplitude and the beam power of the customized laser machine reaches a certain level. Assume that there is no predefined service delivering such a signal but through the composition of the service providing vibration and by consulting the blueprints knowledge base from the standard sensor and a service supporting the special sensor by surveying the beam power, a new composite service satisfying this requirement can automatically been generated and recommended to the customer.

Continuous & Iterative Requirements Validation: Throughout all development phases of the ICP4Life customizer development, validation has been continuously undertaken by ICP4Life industrial partners. To meet their requirements, the customizer is iteratively updated. As certified by industrial partners, this validation step ensures the novelty, validity, usability and applicability of the proposed approach

5. Production Blueprints

In this section we shall present product representation and product line taxonomies to represent the patterns of product portfolios, i.e., the interrelationships among product families, end products (variant products) and building blocks (common modules). This reflects the specific product strategy followed in the ICP4Life Customizer development. To achieve this objective, we employ a *knowledge-driven approach to product design and customization*, i.e., production blueprints, that improves decision-making and product portfolio management thus helping achieve shrinking development times and better product quality. As shown before, production blueprints rely on model-based design techniques to manage and inter-link product data and information (both its content and context), product portfolios and product families, and, in general, help meet the requirements (functional, performance, quality, cost, physical factors, interoperability, time, etc.) of different stakeholders. This information can be collated and put within a broader operational context, providing the basis for manufacturing actionable "intelligence" and a move toward more fact-based decisions.

The production blueprint model is generic in nature but can be easily tailored specifically to the way in which production is performed in a particular sector. To assist manufacturers in developing a wide range of applications, blueprints support well-established software engineering principles such as separation of production concerns, modularization, composition, and reuse. As shown in the right-hand side of Fig. 1, blueprints encapsulate product knowledge from diverse sources including Product Lifecycle Management and Bill of Materials data. They also include supplier profiles and production capabilities, processes and critical manufacturing event descriptions, production schedules and deadlines, and product quality characteristics. As shown in Fig. 1, this knowledge is encapsulated in six inter-connected knowledge-based structures that compartmentalize product and production knowledge and achieve separation of production, i.e., Supplier, Product Family, Product, Service, Production Plan and Quality Assurance blueprints. For brevity, only basic concepts and properties are shown in the blueprint model (right hand-side of Fig. 1). Production blueprints are fully implemented using OWL standard, which is formally founded on DL.

6. Conclusions and Future Work

Product-service systems expose difficult challenges to their developers because they need to balance the advantages of mass production with the need to provide custom-designed products and services to individual customers whose requirements may be understood fully only after a product has been designed and manufactured. To bridge this chasm, we developed a view-based model-driven engineering approach that allows the systematic transformation of an abstract, i.e., implementation and technology-independent product-service model into a tailor-made product-service system. We introduced PoCL, a customer-oriented easy-to-use configuration language, and its technology-oriented counterpart product blueprints.

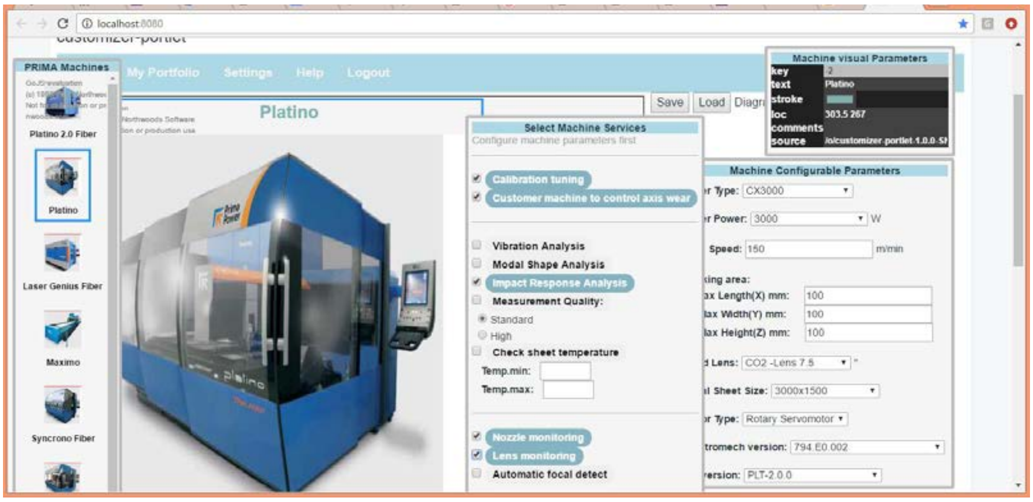

Fig. 3 Screenshot of the Customizer implementation

Both capture a variety of product-service and production-related knowledge distilled from the condensed knowledge about common manufacturing and supply-chain standards and application experiences gained from an FP7 project on flexible manufacturing. A transformation engine that maps PoCL descriptions into product blueprints was also sketched along a concrete use case provided by industry partners. A prototype of a supporting tool suite has been implemented and tested on a use case observed in laser machine industry. Based on the evaluation outcomes of this and three further industrial pilot cases, the initial customisation tool will be iteratively improved and completed.

Acknowledgements

This research is performed in Horizon 2020 project ICP4Life, which is financed by the European Commission under grant agreement number 636862.

References

[1] Production Methods: http://infocheese.com/methodsofproduction.html retrieved 2012-10-26.
[2] Jørgensen K A. Product Configuration and Product Family Modelling. 2009.
[3] Hounshell D A. From the American System to Mass Production, 1800-1932: The Development of Manufacturing Technology in the United States. 1984.
[4] Pine B J. Mass Customization: The New Frontier in Business Competition. 1999.
[5] Gilmore J H and II B J P. The Four Faces of Mass Customization. *Harvard Business Review*, 1997.
[6] Meier H, Völker O and Funke B. Industrial Product-Service Systems (IPS2). *The IJAMT*, 2011, 52(9-12): 1175-1191.
[7] Joergensen S N, Hvilshøj M and Madsen O. Designing modular manufacturing systems using mass customisation theories and methods. *Int. J. Mass Customisation*, 2012, 4(3/4): 171 – 194.
[8] Ulrich K T and Eppinger S D. Product Design and Development. 2004.
[9] Jiao J, Tseng M M, Duffy V G and Lin F. Product family modeling for mass customization. *Computers & Industrial Engineering*, 1998, 35(3–4): 495-498.
[10] MD J and R K. Developing and assessing commonality metrics for product families: a process-based cost-modeling approach. *IEEE Transactions on Engineering Management* 2010, 57: 634-648.
[11] Tian X, Huang L, Jia X and Zhang Z. Exploring Parameterised Process Planning for Mass Customisation. Advanced Design and Manufacture to Gain a Competitive Edge, Yan X-T, Jiang C and Eynard B (eds). Springer London, 2008, pp. 643-652.
[12] Zha X and Sriram R D. Platform-based Product Design and Development: A Knowledge Intensive Support Approach. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, 2006, 19(7): 524-543.
[13] Nomaguchi Y. Proposal of Knowledge Model for Designing Product Architecture and Product Family. In IJCC Workshop 2006
[14] Otto K, Hölttä-Otto K and Simpson T W. Linking 10 Years of Modular Design Research: Alternative Methods And Tool Chain Sequences To Support Product Platform Design. In IDETC/CIE 2013 USA.
[15] Piller F T and Ihl C. Open Innovation with Customers – Foundations, Competences and International Trends: http://internationalmonitoring.de/fileadmin/Downloads/Trendstudien/Piller-Ihl_Open_Innovation_with_Customers.pdf 2009.
[16] Salvador F, de Holan M and Piller F T. Cracking the Code of Mass Customization. *MIT Sloan Management Review* 2009, 50(3): 71 ‐ 78.
[17] Giovannini A. Approach for the rationalisation of product lines variety. In Int. Federation of Automatic Control South Africa, pp. 3280-3291.
[18] T. A, T.C. M and G. C. Product/Service-System Development. In International Conference on Engineering Design ICED'07 2007, France.
[19] Tan A R and McAloone T C. Characteristics of Strategies in Product/Service-System Development. In DESIGN 2006, Croatia, pp. 1435-1442.
[20] Baines T, Lightfoot H W, Evans S,et. al.. State-of-the-art in product-service systems. *Journal of Engineering Manufacture*, 2007, 221(10): 1543-1552.
[21] Papazoglou M P, van den Heuvel W J and Mascolo J E. A Reference Architecture and Knowledge-Based Structures for Smart Manufacturing Networks. *Software, IEEE*, 2015, 32(3): 61-69.
[22] Bock C, Zha X, Suh H-w and Lee J-H. Ontological product modeling for collaborative design. *Adv. Eng. Inform.*, 2010, 24(4): 510-524.
[23] Brambilla M, Cabot J and Wimmer M. Model-Driven Software Engineering in Practice. 2012.
[24] Faltings B and Freuder E C. Configuration - Getting it right. *Special issue of IEEE Intelligent Systems*, 1998, 13(4).
[25] Jørgensen K A. Information Models Representing Product Families. In Proceedings of 6th Workshop on Product Structuring, Denmark.