

Approach for the rationalisation of product lines variety

A. Giovannini^{*,**,***}, A. Aubry^{*,**}, H. Panetto^{*,**}
H. El Haouzi^{*,**}, L. Pierrel^{***}, M. Dassisti^{****}

* CNRS, CRAN UMR 7039, France

** Université de Lorraine, CRAN UMR 7039, Boulevard des Aiguillettes

B.P. 70239 F-54506 Vandœuvre-lès-Nancy, France

*** TRANE SAS, rue des Amériques, 88190 Golbey, France

(e-mail: {antonio.giovannini; alexis.aubry; herve.panetto; hind.el-haouzi}@univ-lorraine.fr)

**** Politecnico di Bari, Viale Japigia 182, 70126, Bari, Italy (michele.dassisti@poliba.it)

Abstract: The product variety management is a key process to deal with the flexibility requested by the mass customisation. In this paper we show that current variety-modelling methods miss a customer representation: without a proper assessment of the customers is not possible to define the product variety that has to be developed to meet the requirements of a customer segment. Here we present an innovative approach to rationalise the product variety, i.e. to link each product variant to the customer profile who needs it. The 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). An overview of customer modelling approaches in the classic product design (non-customisable) is presented. The innovative approach is here developed using system-thinking concepts. A knowledge-based system that uses this approach is designed. Finally the approach is explained using a real industrial case of a quasi-real coil design process.

Keywords: Mass customization, Product variety, Knowledge representation, Knowledge-based system

1. INTRODUCTION: ABOUT MASS CUSTOMISATION AND PRODUCT VARIETY

For a shirt manufacturer, “what if technology made it possible for every one of the five thousand shirts to be customized while on the assembly line [...] produced just as quickly as the five thousand identical shirts, yet at not greater expense?” Starting from these words Davis in (Davis, 1987) introduced for the first time the term mass customisation (MC).

Other definitions can be found in (Tseng et al., 2001) and (Blecker et al., 2004). In these papers, MC is defined as a business approach that aims to fulfil individual customer’s needs having an organizational efficiency of a mass production (MP) company. The MC is explained by comparison also in (Pine II et al., 1993). Authors discuss the change of paradigm needed to move toward MC: if MP or continuous improvement strategies call for low costs and standard goods and services, MC stress on flexibility and quick responsiveness in order to cope with the products customisation.

Another recently introduced concept is the Mass Craftsmanship (MCMS) that is a hybrid form of pure MC and mass personalization, since it is possible to address both wide market and unique requirements. An interesting application in the furniture sector is discussed in (Dassisti et al., 2012), which is mainly similar to that addressed in this paper.

From these definitions, it is clear now the big deal in MC is to timely react to the complexity deriving from the variety of the customer domain. This variety impacts directly the number of products to be manufactured and consequently, the complexity of the manufacturing

system and of the planning and scheduling processes (Blecker et al., 2004).

Most of the related works cope with these effects of the product variety. Complementarily, in this paper, the aim is to reason about the origin of this variety, in order to define the right level of customisation to propose, i.e. the number of product variants. For a better understanding, the example of a shoes seller who offered $3 \cdot 10^{21}$ variants of its shoes is described in (Piller 2003). All together, these shoes can cover 7,000 times the surface of the earth.

In the next section, we formalise the modelling requirements to rationalise the product variety. In the section 3 current variety modelling methods are presented: here, defining a gap between these systems and the customer domain. To understand how to bridge this gap, in the section 4, we focus on the methods typically employed for designing non-customisable products: here the aim is to discuss which method is the more appropriate for expressing formally the variety in the customer domain. Starting from methods discussed in the section 4. A conceptual model that applies a System thinking view on the product variety definition is presented in the section 5. Using this conceptual model as knowledge model, a knowledge-based system is designed in section 6. Finally, section 7 is dedicated to present an application of an industrial case. In the last section, results and future perspectives are discussed.

2. MODELING REQUIREMENTS FOR VARIETY RATIONALISATION

In order to identify the modelling requirements to rationalise the product variety, let us first define two generic sets X and

Y . The set X represents a targeted customer segment. An element of X is a vector of features in the customer domain expressing one customer. The set Y represents the set of product variants defined for the customer segment: an element of Y is a vector of features in the engineering domain expressing one product variant (see Fig. 1a). The function g is the product variety definition function and it has Y as domain and X as codomain. One product variant can correspond to more than one customer. This means that two different element of Y can be related by g to the same element of X . In other words, the product variety definition function g can be multivalued (see Fig. 1b).

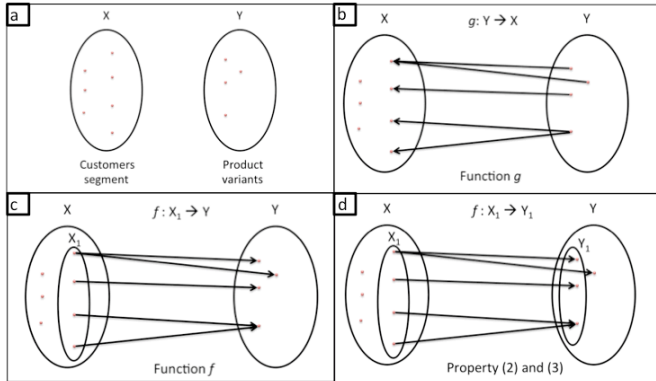


Fig. 1 – how the relationships between the customers and the product variants allows to define the right product variety

If X_1 is the image of the function g , we can define a function f that has X_1 as domain and Y as codomain. The function f is the configuration function that allows to link the customers for which a product is feasible (set X_1) with the product variants in Y . The configuration function f is defined as the inverse function of the multivalued definition function g (represented also in Fig. 1c).

In order to avoid unnecessary development costs, an engineer should design no more than one variant for each customer in the customers segment: these variants have the same added value for the customer, but can represent different resources consumption for the customizer. This requirement can be formalised by the following property of the configuration function f :

$$\forall x_1, x_2 \in X, f(x_1) \neq f(x_2) \rightarrow x_1 \neq x_2 \quad (1)$$

There exists more than one subset of Y in which f respects this property: the choice of the best one should be done according criteria from economics, sustainability and so on. A subset of Y , in which the configuration function respects the defined property (1), is called Y_1 . If a function f respects the property (1), a customer cannot be related to more than one product variant. In other words, the function f defined from X_1 to Y_1 is a single-valued function (see Fig. 1d).

If f is the inverse function of g and respects (1) then we can infer that f covers completely Y_1 : a product variant that is not related with any customer cannot exist. This means that an engineer is not involved in the development of a product variant that will not be sold to any customers.

Starting from these definitions, the set Y_1 represents the right product variety for the customer variety X_1 . From the whole customers segment X , the function g allows to identify the set X_1 representing the customers that is possible to satisfy. A configuration system that expresses the right level of product variety is a system that uses the configuration function f as in (1) to allow each customer described in X_1 to configure only one product variant in Y_1 .

In the next section, an overview of works on variety modelling, especially for product configuration, is presented. The aim is to verify if the state of the art includes solutions able to model the set X_1 , f and consequently the set Y_1 (i.e. formalisation of the right product variety).

3. CURRENT VARIETY MODELING METHODS

The models that formalise the variety (i.e. set Y) in MC are usually named *product family* (Hong et al., 2008), *configurable product model* (Aldanondo et al., 2008) or *product line* (Pohl et al., 2005; Mazo et al., 2012). In this paper we refer to them as product lines (PLs). A PL is the model that expresses the product variety, i.e. set Y , that a company offers to a given customer segment in order to fulfil the needs of customers in the segment.

A PL is the base model for every configuration system, used by the customer to select one product variant starting from its needs. In all configuration systems, the customer describes his needs giving some values and preferences to attributes related to the PL variety. An attribute is defined as “an inherent property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means” (ISO/IEC/IEEE, 2007). Starting from the attributes values, the configuration system is able to define the product variants into the PL variety that are related to the customer needs. For instance, the range of values for the attribute screen size can represent the product variety of a laptop PL.

In the following figure from (Jiao et al., 2007), a complete view of the PLs design and development is shown. The authors define there four main processes for translating the customer needs into design, manufacturing and logistic requirements:

- *Product definition* – how to offer the right variety to the right target market?
- *Product design* – definition of parameters or functional modules (PL attributes) for deriving from the PL the product that fulfils the customer needs
- *Process design* – the management of process variety with process platforms and process configuration methods
- *Supply chain design* – the management of variety on sourcing, manufacturing and distribution strategies.

Here, *customer needs* represents the customer’s description of the ideal product (e.g. cute, easy to use). *Functional requirements* concern the features owned by the product, seen as a whole (e.g. colour, size). *Design parameters* are features characterising product modules, subassemblies and components. Process and logistic variables deal with production and supply chain (not the focus of this paper).

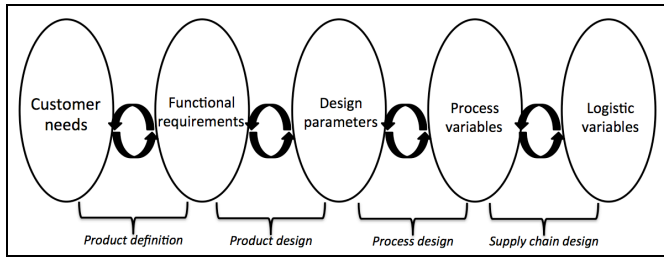


Fig. 2 – An holistic view of PL design and development, adapted from (Jiao et al., 2007)

In this section, an overview on PLs modelling solutions and related configuration systems is presented. In Tab. 1, related works are classified according to the above-defined terms related to customer needs, functional requirements and design parameters. The analysed research works are classified according to the semantics of the attributes that expresses the PL variety regarding the customer in current configuration systems. Essentially, we consider here research works that model concepts related to the customer, e.g. customer needs, customer profiles. The analysis of the solutions developed so far allows us to identify a gap between the semantics of these concepts in the cited scientific works and the customer domain representing the set X , as formalised in the previous section. This gap prevents the presented configurations system to identify a function f and therefore the set Y_1 representing the right product variety.

Product configuration from design parameters

In order to configure a product starting from design parameters, the customer is asked to know how components interacts and what the final product features are. Here the required knowledge is technical. For instance, in configuration processes of laptops, the variants description is typically based on memory size (RAM), processor speed, and so on. In each case, the customer has to have a good knowledge about a computer architecture even if its needs concern the ability of the laptop to go on internet, to write some documents, to watch videos, to play videogames and so on.

In this kind of systems, a clear definition of the customer needs as required for the definition of X is not expressed. This lack in the definition of X hinders formalising the configuration function for linking the customers segment with the right level of product variety, i.e. Y_1 . The effect on the product variety is that, two customers having the same needs, and this representing only one element of the X set, can configure two different product variants, as a consequence of their knowledge about the computer architecture. This violates the property (1) causing a lack of control on the level of product variety proposed to the customers.

Product configuration from functional requirements

In configuration systems based on attributes described as functional requirement, the customer would be asked to configure the product starting from a holistic view of the product. In this case the customer has to know how the product interacts with its environment to fulfil his needs. For instance, in a heating or cooling system, the power of the system is a property and is not the only one directly related to

a room temperature value that can represents a real customer need. In order to reach a temperature target we need to know about room usage, isolation and so on.

Also in this situation, two customers with the same needs, represented as only one element of the set X (see section 2), can configure two different product variants, according to their experience about the usage of these kinds of systems. As for the systems in the previous subsection, configuration systems based on functional requirements cannot express univocally the right product variety.

Product configuration from customer needs

In (Helo et al., 2010), the customizer uses market research. In this case, the manufacturer designs the products and after looks for potential customers for defining a variety in the customer domain. Here, the product variety is defined before defining the customers segment. Doing so it is not possible to guarantee that each product variant meets the requirements of at least one customer in the targeted segment. Moreover, even if the customer does not express his needs as design parameters or functional requirements, he answers also questions about the product (e.g. “which part of the truck do you like most?”). Therefore, also in this case a formalisation of the customers segment, as in X , is not feasible.

3.1. *The effect of the gap: missing a consistent model of the customer segment*

As a consequence of the recognised gap between the current configuration systems attributes and the customer domain, as defined by the set X , in current configuration systems, the customer is asked for describing something about the product. The effects of this lack of the definition of the set X are shown in the Fig. 3.

Without a formal definition of a set X representing the customer variety, a formalisation of a unique configuration function f , as described above, is never performed. In other words, each customer follows a different f based on his knowledge; therefore the selection of the best variants (according f_1 in Fig. 3) cannot be performed on the engineering side. The absence of a unique formalised f implies that a customer, element of X , can also configure (according f_2 in Fig. 3) a product variant that does not represent the best choice for the enterprise (see the definition of Y_1). In the worst case (according f_3 in Fig. 3), the customer can select something not included in the available product variety. In summary, without a consistent formalisation of X , a definition of a set Y_1 representing the right product variety is not possible.

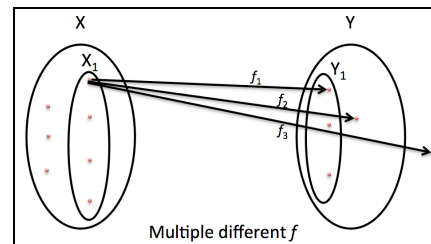


Fig. 3 – Effects of the gap between configuration systems and the definition of X

The gap between the representation of the product variety and the customer domain - that is not covered by current configuration systems leads to build configuration systems that cannot express the right level of product variety. In order to bridge this gap, we need to understand how to define and formalise the customer variety and what is the knowledge needed for the definition and configuration functions and the set Y_1 . In the next section an overview on marketing and engineering methods for the non-customisable *product definition* processes is presented. The aim is to understand how the link between the customer domain and the product is performed for non-customisable product and then apply this solution to the variety modelling.

4. CUSTOMER MODELING FOR NON-CUSTOMISABLE PRODUCTS DESIGN

Typically the *product definition* process, for the PLs as well as for simple products, is discussed in the marketing domain and during some engineering phases, i.e. the early design stages.

4.1. Marketing domain

Usually, the solutions to cope with the *product definition* are in the domain of marketing. We can identify two main approaches: a “attribute-based” and a product “alternative-based” product choice.

The first group is mainly based on the conjoint analysis (Green et al., 1978). This technique represents a milestone: this is an attribute-based method that aims to study the customer behaviour; the customer preferences on product attributes are analysed in order to derive a utility function to be maximised during the design stage. An application can be found in (Balakrishnan et al., 1996): here authors use a genetic algorithm to elaborate the conjoint analysis data for design optimisation.

Another application is suggested in (Tseng et al., 1998), where the *designed by customers* method is presented. Here authors describe how to integrate in an engineering design the conjoint analysis results. In (Krishnan et al., 2001), authors define a *consumer valuation* that is calculated from the consumer utility per performance unit. In (Hong et al., 2008), authors represent the formalisation as a customer satisfaction optimization model for solving a configuration problem. A genetic algorithm is deployed for finding a solution. Graphs are used for linking product components features with customer needs. To the second group of methods belongs the decision-based design (Hazelrigg, 1998). For example, in (Wassenaar et al., 2001), authors compare the discrete choice analysis (DCA) with the conjoint analysis. In the former, the customer choice is directly related to product alternatives. The hypothesis here is that for choosing which product to buy is what customer does best. Actually, in (Huffman et al., 1998), authors present experimental results on the customers’ satisfaction in front of product variety. They show how too much variety can be frustrating for the customers and an attribute-based selection is to prefer to an alternative-based, because the first one can simplify the product variety *vis-à-vis* the customer.

In all described marketing methods the analysis of the customer preferences starts with the evaluation of solutions, represented by product attributes or directly by product alternatives. In this way, the customer is asked to talk about the product. Therefore a formalisation of a unique configuration function f , as described above, is never performed. In other words, each customer follows a different f based on his knowledge.

The absence of a unique formalised f implies a lack in the characterisation of the customers segment, i.e. the set X . In summary, the usage of this type of methods for the PL definition lead to the cases discussed in the section 3;

Tab. 1—overview on configuration systems

	<i>Customer needs</i>	<i>Functional requirements</i>	<i>Design Parameters</i>	<i>Deployed configuration method</i>
(Aldanondo et al., 2008; Mazo et al., 2012; Yang et al., 2012)			x	Constraint-based approach
(Du et al., 2001; Janitza et al., 2003; Qin et al., 2010)		x		Mapping approach
(Gao et al., 2003; Liu et al., 2003)		x		Ontology mapping approach
(Helo et al., 2010)	x			Ontology mapping approach
(Hong et al., 2008)		x		Genetic programming
(Inakoshi et al., 2001)			x	Case-based reasoning & constraint-based approach
(Li et al., 2006)		x		Genetic algorithm
(Wang et al., 2011)			x	Ontology mapping
(Xie et al., 2005)		x		Constraint-based approach
(Zhou et al., 2008)		x		Constraint-based approach & genetic algorithm
(Zhu et al., 2008)		x		Fuzzy decision making approach

without a consistent formalisation of X , a definition of a set Y_1 representing the right product variety is not possible.

4.2. Engineering domain

In this section, some of most known engineering design solutions are briefly introduced. Most of those are related to the concepts of product and process quality.

One of the best approach oriented to close the gap between customer domain and engineering domain is the design for 6S (Ferryanto, 2007). One of the basic tool for this is the Quality Function Deployment (QFD) method (Ferryanto, 2007). The application of QFD for product design provides an effective methodology for the recursive translation of the voice of customer into customer needs, so as to facilitate their technical specification. QFD allows (1) to analyse the requirements demanded by customers in their own words, (2) to translate the relationships between these and product quality properties, (3) to convert customer requirements into technical ones (Akao, 2004).

The lean production (LP) philosophy is based on the experiences of the Toyota Production System (TPS) (Shah et al., 2007). The LP focus is on the waste minimisation. According authors in (Spear et al., 1999), LP is based on four main rules: (1) all work shall be highly specified as to content, sequence, timing, and outcome; (2) every customer-supplier connection must be direct, and there must be an unambiguous yes-or-no way to send requests and receive responses; (3) the pathway for every product and service must be simple and direct; (4) any improvement must be made in accordance with a scientific method, at the lowest possible level in the organization. Also the lean philosophy, like for design for 6S, has a branch that focuses on the product development. In (Morgan et al., 2006), authors define 13 principles of the lean product development. The first one concerns the characterisation of a customer-defined value for distinguishing the value-added features from the waste. According authors in (Liker et al., 2011), the best way to deploy this principle is to go beyond the QFD performing a *Requirement Engineering* (RE) process.

Based on theoretical principles from the General Systems Theory (Von Bertalanffy, 1956), in last years a new engineering discipline has been developed: System Engineering (SE). The focus of SE is on the design of complex systems that usually require multidisciplinary exchanges of knowledge and expertise. Main system models are (Pyster et al., 2012):

1. *The System requirements model* that contains the required system features in order to fulfil customer needs; this model is independent from solution alternatives and from technology;
2. *The Logical architecture model* that describes how the system functions, i.e. its behaviour for the requirements fulfilment; this model is independent from technology
3. *The Physical architecture model* that describes the set of elements that perform functions; this model is related to technologies.

The process that performs a translation from customer requirements to system requirements is named RE. The main purpose of RE “is to produce a set of system requirements which, as far as possible, is complete, consistent, relevant and

reflect what the customer actually wants” (Sommerville et al., 1997).

All methods described in this section perform an analysis of the customer domain. In QFD, the object described by the “voice of the customer” is not specified. This means that the customer can express some requirements about the product features as well as about his own environment (in which the product has to work) or the required effects in this environment. Despite requirements clustering may help to solve this problem, the application of the QFD for PL definition still lacks of precision: a model representing a variety of the voice of the customer is not possible. Therefore the first formalisation of the variety is only possible on the technical side, according to the product description. This leads to a similar position with the discussed for marketing: an abstraction of the customer needs to obtain a characterisation of the set X is not possible. A missing definition of X does not allow assessing the proposed product variety. As mentioned in (Liker et al., 2011), RE allows going much further than QFD. Actually, in SE a better definition of customer needs, customer requirements and system requirements are provided. Already in our initial work (Giovannini et al., 2013), we identified the gap according SE, describing the lack of precision in the definition of customer needs for current configuration systems.

In the next section, an application of these definitions on the variety modelling is discussed. The aim is to identify how to formalise the variety in the customer domain (set X) and in the engineering domain (set Y) and what is the nature of the knowledge formalised by the functions f and g .

5. APPLYING THE SE CONCEPTS TO THE VARIETY MODELING

The proposed approach relates the PL concept to the definitions of System Engineering (Pyster et al., 2012) and concepts of system thinking (e.g.(Jackson et al., 1995)); the aim is to provide a formal definition of the sets and functions described in the section 2.

5.1. Basic terms from SE

The current section presents the SE concepts that relate, by means of their semantics, the person (generalisation of a customer – element of X) with the system requirements (elements of the product variety). The following definitions served as a basis for building the conceptual diagram in Fig. 5a.

According to SE terminology, the interaction between the *person* and its *environment* is moved by a *purpose*. The *purpose* represents “what the system is for, and why the different stakeholders are willing to participate in the system lifecycle” (Blockley et al., 2000). The *person* who participates to this relationships is a *stakeholder*, where stakeholder is defined as “a party having a right, share or claim in a system or in its possession of characteristics that meets that party's needs and expectations” (ISO/IEC, 2008). The definition of *purpose* matches the definition of *real needs* in (Faisandier, 2013): here the author defined as real needs those needs “that lie behind any perceived needs (defined below) and they are conditioned by the context in

which people live”. In the same paper, the author presents a diagram of the needs lifecycle (Fig 4) that shows the transformations from real needs to a set of consistent requirements. In this figure, real needs are related to *stakeholder requirements* by *perceived needs*, expressed needs and retained needs:

- *perceived needs* “are based on a person’s awareness that something is wrong, that there is a lack of something, that there is something that could improve a situation, or that there are business, investment, or market opportunities”
- expressed needs “originate from perceived needs in the form of generic actions or constraints, and are typically prioritized”;
- retained needs “are selected from the expressed needs. The selection process uses the prioritization of expressed needs to achieve something or make solutions feasible “.

The retained needs represent the stakeholder intentions that still “do not serve as stakeholder requirements, since they often lack of definition, analysis, and possibly consistency and feasibility [...] RE leads stakeholders from those initial intentions to structure more formal stakeholder requirement statements” (ISO/IEC/IEEE, 2011). *Stakeholders requirements* are “the requirements for a system that can provide the services needed by users and other stakeholders in a defined environment” (ISO/IEC, 2008). According to authors in (Jackson et al., 1995) the *stakeholder requirements* have to constraint the environment of the system, stating the expected effect on the environment caused by the interaction with the system. *Stakeholder requirements* are transformed by a RE process into system requirements. *System requirements* are defined as “the requirements for a system that can provide the services needed by users and other stakeholders in a defined environment” (ISO/IEC, 2008). A detailed analysis of the process for transforming *stakeholders requirements* into *system requirements* can be found in (Bouffaron et al., 2012).

System requirements are collected in a *system requirement model* (SRM), as defined in the previous section. As *system requirements*, also *stakeholder requirements* are collected in an appropriate fashion (*stakeholder requirements model* - RM). This environment is different from the one of person i.e. the *person* can be part of the *system environment*. this latter includes “anything affecting a subject system or affected by a subject system through interactions with it, or anything sharing an interpretation of interactions with a subject system” (IEEE, 2002). The system here is the system whose life cycle is under consideration, named the *System-of-Interest* (SoI) (ISO/IEC, 2008).

Fig.5(a). shows the conceptual model expressing the main definitions from the previous section. Notice that the *stakeholder requirements* are the first statements really structured and consistent, in the customer domain.

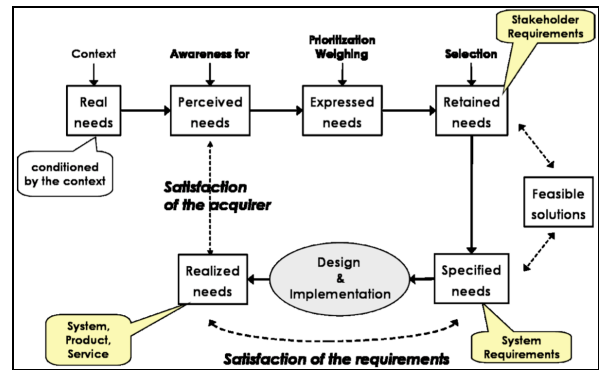


Fig. 4 – Needs lifecycle (Faisandier, 2013)

5.2. The conceptual model for representing the right variety

Customer and product variety

In order to define the sets X and Y , let us now consider a *SoI* referring to a PL so that it has to describe a certain variety interacting with an environment now related to a *customer segment* (see Fig. 5b). A *segment environment* is related to more than one RM. Similarly the *SoI* is related to more than one SRM. Since a requirement is unambiguous and measurable (ISO/IEC, 2007) and a SRM models consistently a set of system requirements, a SRM can represent an element of the set Y . As for PLs (see section 3), a set of attributes can describe the variety and similarities between different SRMs: this variety can formalise the product variety (i.e. set Y) for the customers segment.

As analogy of SRM, the values related to a set of attributes describing the variety between the RMs will represent the customer variety, i.e. the set X . This implies that the customer variety is defined by the characterisation of the system environment and the effects that the system has to provide on its environment. The difference between this definition of X_1 and the definition of the set X concerns the feasibility. The *stakeholder requirements* are related to a feasible system (ISO/IEC/IEEE, 2011) therefore every RM has a correspondent SRM.

The PL definition function

Since the *stakeholder requirements* constraint the environment and the system requirements specify the SoI, the knowledge behind the PL definition function g (and consequently f as inverse of g) allows relating attributes concerning the system environment and attributes specifying the SoI. Therefore it represents the knowledge of the interactions between the SoI and its environment. Since the function g is multivalued, more than one SRM can be found for a RM.

In this section, we have defined the set X , X_1 , Y and the definition function g on the basis of terms from SE.

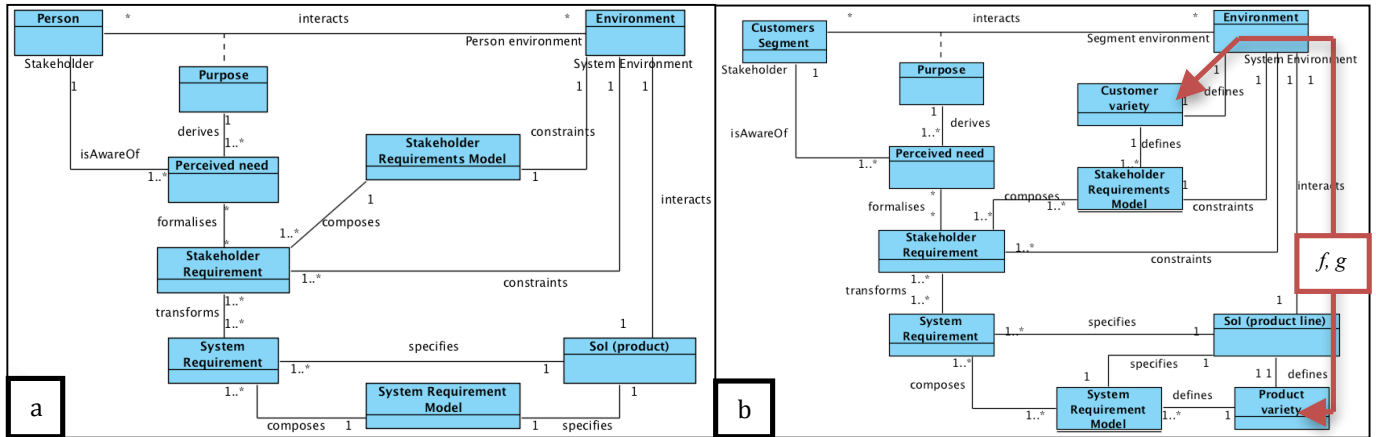


Fig. 5 – Conceptual models of discussed SE terms (a) and PL variety (b)

6. KBS FOR PRODUCT LINES DEFINITION AND RATIONALISATION

In this section, we show how to involve the knowledge before defined in the definition or rationalisation of a PL. At this purpose, a knowledge base that uses the concepts just described is used as a core for a Knowledge Base System (KBS). The KBS inferences support the definition of the configuration function f and therefore of the set Y_1 , representing the right product variety for the PL.

The first subsection of this paragraph discusses the KB formalisation by means of an ontology. The second subsection describes how the KBS support the definition and the rationalisation of a PL.

6.1. From the conceptual diagram to the knowledge base

Starting from the conceptual diagram just discussed (Fig. 5b), we define an ontology as the core of a KBS for the PL definition and rationalisation. The ontology has been formalised using a description logic (DLLite) (Baader, 2003). DLLite is based on the theory of sets and guarantees the decidability; therefore a verification of the consistency of the modelled artefact can be performed. Moreover DLLite is based on the open world assumption (OWA), thus the knowledge base can be easily extended. In order to translate the conceptual diagram into an ontology we used the mapping rules defined in (Atkinson et al., 2008).

The key knowledge formalised in the ontology is about the function g , i.e. about the relationships between attributes representing customer variety and attributes representing product variety. As discussed in the previous section, this implies that the function g describes the interactions between the SoI and its environment.

In the ontology representing the KB, a class concerning the knowledge just described has been formalised: the class interaction. In this class, individuals describe the mathematical knowledge for calculating values of a required attribute, starting from values of the other attributes related to the interaction. The knowledge about interactions may come from physics, chemistry, experiments, personal experience, etc.. This class relates attributes describing the environmental expected effects and constraints with the system attributes (Fig. 6).

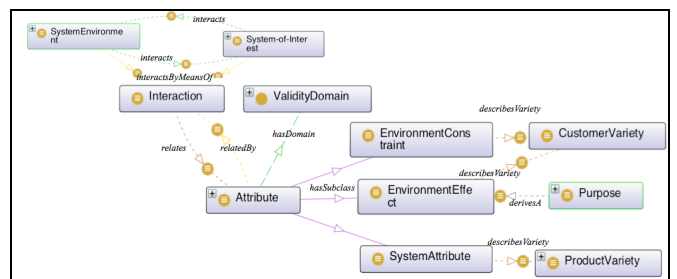


Fig. 6 – excerpt of the KB formalised in Protégé (Protégé)

The semantic difference between environmental expected effect and environmental constraints concerns the definition of the purpose shown in the previous section. Actually, a system purpose emerges whenever an unsatisfactory interaction between a person and that person's environment exists. Consequently a purpose represents something that the system, working in the system environment, has to change. This means that expected effects on the environment are derived starting from a purpose. On the other hand, environmental constraints represent something that the system, working in its environment, has to take into account as some perturbations.

System attributes characterise the product variety. They represent what the system should do to get the environmental expected effects, respecting the environmental constraints.

For instance, we can consider temperature of a room as an environmental expected effect, the room usage and size as environmental constraints and the cooling system power as a system attribute. Interactions in the KB have to describe the mathematical relationships between the values of these attributes.

Environmental effects, constraints and system attributes are all related to the class *attribute* of the ontology. As discussed in the previous sections, every attribute is a measurable property. Every attribute can be related to different interactions and this relationship is characterised by a domain of validity. Returning to the previous example: if the knowledge contained in the KB is limited to temperature of 24, 25 and 26°C, these three values represent the domain of validity of the attribute along the considered interaction. In other words, if the system attribute is the power of a heating

system, for defined room characteristics, we know only this information related to maintaining the room temperature at 24, 25 and 26°C.

6.2. Usage scenario for the KBS

Fig. 7 shows the KBS usage scenario. The user selecting the purpose of the SoI in the system performs the first activity. As shown in the conceptual diagram, the purpose for a *customer segment* is considered as unique. An example of purpose can be “maintain a good air quality in the room”.

Algorithm 1 – interactions retrieval

The KBS retrieves the knowledge about the introduced purpose in the KB. If at least one correspondence is found, the algorithm1 checks for environment effects derived by the purpose. This algorithm builds a sort of tree of interactions. When an attribute is related to an interaction, the algorithm identifies the other attributes involved in the interaction. Then, for each identified attribute, the algorithm retrieves interactions to which it is related and other attributes are identified. These tasks are iterated until an attribute is an environment constraint or, in the case of system attributes and environment effects, if there are no other interactions related to it. The result of the algorithm is a tree that has as roots the environment effects related to the purpose and as leaves the system attributes and the environmental constraints. Every attribute is related to a validity domain for the related interaction. According the example above, the algorithm would infer attributes on customer variety such as desired temperature, size of the room, usage of the room and so on. On the PL side, possible inferred attributes could be the power of the system, air-filtering features and so on.

When the tree includes alternative paths, more than one PL impacts the same purpose. This implies that a PL is defined by the attributes on the leaves of the tree. In other words, two PL can have the same purpose but cannot be addressed to the same customers segment, i.e. one PL for each customers segment. Returning to the example above, two different PL can affect the air quality in a room, but they can have different interfacing requirements with the environment, e.g. solar heating system, hydraulic system or cogeneration. In this case the customer variety is characterised by attributes that are completely different.

On one tree path built by the algorithm 1, the interactions represent the PL definition function g . The set of system attributes and their domain validity describe the variety in the product domain Y . The set of attributes for environment effects and constraints and their domain of validity define the variety in the customer domain, i.e. set X_1 . One element of X_1 or Y is a vector that has as component a value for each attribute describing the variety. The multivalued function g transforms a vector of system attributes representing a product variant into a vector of environmental effects and/or constraints representing the customer variety.

The inference algorithm 1 is external to the ontology. This is due to the OWA. Actually we need to make design decision based only on the knowledge available at a defined time, i.e. condition on the end of iteration for the tree construction. For

this kind of inferences, the closed world assumption (CWA) is required. An external algorithm manipulating the ontology can guarantee the CWA.

Algorithm 2 – PL definition

Starting from a customer segment characterised by certain features, the user should select a PL starting from the customer variety expressed by, i.e. environmental effects, constraints and related validity domains. The latter limits the environments in which the product variants can work for the defined purpose. For instance, if the attribute “room size” has an associated range of validity, this means that we are not able to propose variants that meet the purpose for room sizes outside of that range.

Then the user can further characterise the customer variety defining the values for each attribute, i.e. environment effects and/or constraints. Only values inside the validity domains can be introduced. For instance, if it is known that there exists a minimum room size legally defined, the user can limit the domain validity of the related attribute. It is important to highlight that these characteristics allows determining a finite set representing the customer segment variety. Obviously, if the customer set is unbounded then an assessment of a limited PL variety is not feasible.

Once the values for attributes of a customer variety are introduced, the algorithm 2 uses the interactions retrieved by the algorithm 1 for calculating constraints in terms of system attributes. These attributes limit the border of a space that represents the variety that the PL has to express in order to meet all requirements in the defined customer segment. Each system attribute and related validity domains represent respectively the dimensions and the limits of the space. Moreover, a validity domain for environmental effects and constraints can be characterised by a tolerance: the latter can allow making discrete a continuous interval; in each case, the space defined by the system attributes should be discrete. Notice that, on the product variety, a variant represents a physical object; therefore a product variety defined by continuous intervals would be nonsense.

Even if limited, this space still does not represent the portion of the set Y_1 defining the right product variety. Actually, Y_1 represents the codomain and image (property 3) of the configuration function f , where f respects the property (2).

To respect the property (2), the configuration function has to relate each customer with only one product variant. This implies that, in order to respect the property, only one product variant should be chosen. Since for the customer all these variants are equally fulfilling his needs, the most reasonable decision is to choose the best product variant for each identified environmental effect: for instance, an optimisation criteria could represent the lowest resource consumption for the development and the manufacturing of the related product. If the property (2) is respected, the set of product variants defined by system attributes and values represents the set Y_1 , expressing the right product variety.

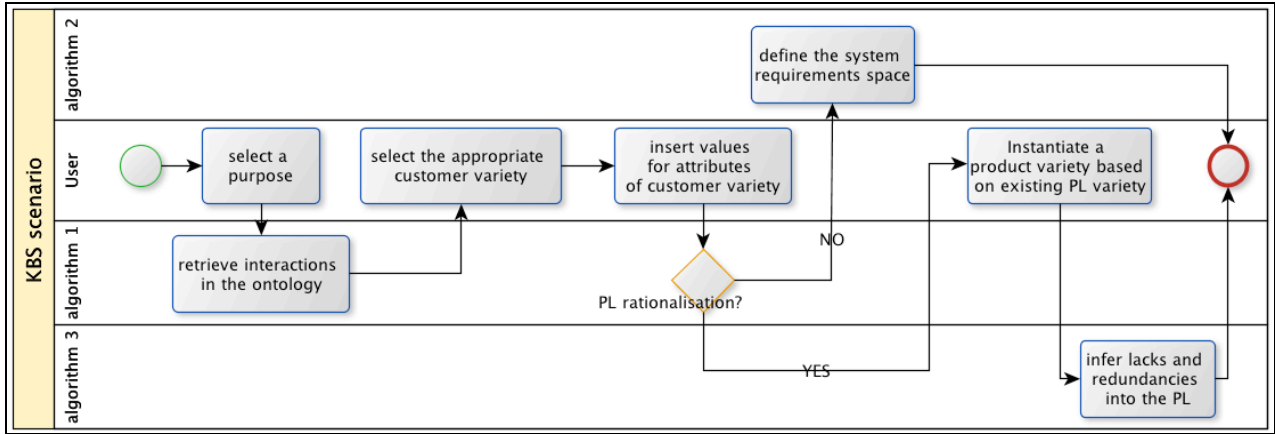


Fig. 7 – Usage scenario for the KBS for PL definition and rationalisation

Algorithm 3 – PL rationalisation

When the aim is the rationalisation of an existing PL, the results of the algorithm 1 are used by another algorithm. This time, the user introduces values also for system attributes. These values represent the variety expressed by the existing product line. The algorithm 3 uses these values and the interaction in the tree to infer the coverage of the PL on the defined customer segment. The ranges of values for the attributes describing the customer variety define this coverage. Comparing these ranges with the defined customer segment, this algorithm allows to highlight lacks, redundancies and/or excess of variety in an existing PL: a lack represents a part of the defined customer segment that is not covered by the PL; a redundancy in the PL means that it is possible to identify more than one variant for a distinct customer; an excess of variety represents a part of the PL variety that cover customers outside the customer segment defined by the user.

7. CASE STUDY

In this section, an application of the proposed approach is presented. The aim of this case study is to show how the KBS should handle domain knowledge in order to rationalise the product variety.

The case study presented is typical of the major part of companies producing mature products, where the revenue is strongly dependent on scale economies more than on innovation lips. In such a kind of scenario, the careful definition of tacit and explicit customer requirements became a critical asset for revenue margins. The challenge for any company in mature market scenario is to increase the worthiness of money spent by customer, by providing satisfaction of the tacit customer's expectations in terms of product quality and reliability, while maintaining a high level of profitability. The mass craftsmanship paradigm is here thus implemented by applying the methodology proposed to solve this apparent contradiction.

KBS scenario

The KBS should allow the selection of a purpose for the PL. The analysed PL has as purpose the “room air-cooling”. In the KB, this purpose is related to some attributes representing the desired environmental effects for cooling the room, i.e. the room temperature. The algorithm 1 defines a tree of

interactions for this attribute. The retrieved equations are the following:

$$q_{td} = U_0 F_s A_a N_r \frac{(t_{a1} - t_{r2}) - (t_{a2} - t_{r1})}{\ln [(t_{a1} - t_{r2}) / (t_{a2} - t_{r1})]}$$

$$A_a = Q / V_a$$

$$q_{td} = P_a = P_r = P_s$$

$$F_s = A_0 / A_a N_r$$

$$U_0 = \frac{1}{(1/\eta f_a) + (B/f_r)}$$

$$P_a = 0.018 A_a V_a (t_{a1} - t_{a2})$$

$$P_r = \rho_r Q c_r (t_{r1} - t_{r2})$$

$$Q^2 k = P_{st}$$

$$B = A_a / A_i$$

Notice that more than one attribute can be related to the selected purpose; this should not affect the algorithms effectiveness. The tree has no alternative path and the attributes on the customer side (i.e. environmental effects and constraints) are: the room temperature (t_{a1}), the fluid type (ρ_r, c_r), and input output temperatures (t_{r1}, t_{r2}), the static pressure (P_{st}) of the duct and the power (P_s) to be provided to the room depending on building features and the room usage. The attributes on the engineering side (i.e. system attributes) are the fan P_{st} /air flow characteristic (k), the air velocity (V_a), the external coil surface (A_0), the internal coil surface (A_i), the number of coil rows (N_r), the coil air heat exchange (f_a), the coil fluid heat exchange (f_r), the fin efficiency (η), where the last three properties are the result of the interaction of the water and air flows with the coil geometry.

The user has to characterise the customer segment introducing a range of values for each environmental constraint and effect. For instance we can consider 21°-25° C for the room temperature; only water as cooling vector (fluid)

type; 5-12 °C for fluid input temperature and 10-17°C for output fluid temperature; 0-60 Pa for environment static pressure; 0.5-10kW for sensible power.

Starting from the retrieved equations and ranges of values, the algorithm 2 should limit the values of the system attributes. In this case, considering the attributes range for static pressure (P_s) (0-60 Pa) range of values, it is possible to add two more constraints to the retrieved equation $Q^2k = P_{st}$, where Q is the airflow rate and k is the corrective constant depending on the fan type. Let us consider for a moment this interaction as the only one in the tree. This means that the environmental constraint can be fulfilled by couples of values for the airflow and the fan type. From the customer point of view, the choice of two couples of values generating the same static pressure makes no differences. Therefore engineers have to choose only the best one (e.g. on the basis of economic reasons) in order to preserve the property (2), i.e. one variant for each customer. Defining a tolerance values for the environment constraints and effects (e.g. ± 0.1 for t_{a1}), the constraints can support the definition of discrete intervals of system attributes and consequently a finite number of product variants. Variants should be selected defining some optimisation criteria to the defined constraints. The algorithm 3 for the PL rationalisation requires also the system attributes to be filled. Here the user has to describe the existing PL variety according to the inferred system attributes. The algorithm 3 should use the interactions retrieved by the algorithm 1 for projecting the PL variety on the customer segment. The algorithm should highlight lacks, excess and/or redundancies in terms of values for the static pressure, the room temperature and so on.

The object of the case study is inspired by the works on the Trane fan-coils PL, named UniTrane™ FCD (Fig. 8). Trane is a company that designs, develops and commercialises heating and cooling systems. For confidentiality reasons, the knowledge we use to instantiate the ontology is not coming from the Trane PL, but the domain knowledge in (American Society of Heating, 2012) is used to show how the KBS should perform in this scenario. Fan-coils, as defined in their name, are products made by two main components: the *coil* is connected to the pipe circuit of the building; generating a resistance at the fluid flow, the *coil* transfers heating or cooling power from the fluid in the pipe to the air flow on the internal part of the fan-coil unit; a *fan*, connected to a suction and an injection duct, feeds this flow and allows the power transfer between the *coil* and the room that has to be heated or chilled.



Fig. 8 – Picture showing a variant of the Trane fan-coils PL

Knowledge representation in the ontology

The knowledge derived from (American Society of Heating, 2012) is typically formalised as mathematic formulas, describing physics principles and experimental relationships. An examples is the following:

$$q_{td} = U_0 F_s A_a N_r \frac{(t_{a1} - t_{r2}) - (t_{a2} - t_{r1})}{\ln [(t_{a1} - t_{r2}) / (t_{a2} - t_{r1})]}$$

where q_{td} is the sensible cooling capacity of a coil, U_0 is the overall coefficient of heat transfer for sensible cooling (without dehumidification), F_s is a coil shape coefficient, A_a is the coil face area, N_r is the coil rows number, t_{a1} and t_{a2} are the input and output air temperatures, t_{r1} and t_{r2} are the input and output fluid temperatures.

In the ontology, this is an individual belonging to the class *interaction*. All variables in the formulas represent the attributes related by the interaction. Each attribute is marked as being an environmental constraint (customer variety), an environmental effect (customer variety depending on the purpose) or a system attribute (product variety). In the formula above: q_{td} , U_0 , F_s , A_a and N_r are system attributes, t_{a1} is an environmental effect (approximation of the room temperature), t_{a2} is a system attribute and the fluid temperatures (t_{r1} , t_{r2}) are environmental constraints.

8. CONCLUSIONS AND PERSPECTIVES

In this paper an approach for the rationalisation of the product variety has been presented. Initially, a formal definition of the requirements for the rationalisation is proposed. Related concepts have been identified by means of a conceptual model based on system engineering (SE) terminology: customer has been represented as environment and the relationship with product as an interaction between the system and its environment. Concepts have been formalised and extended in an ontology for representing PL definition knowledge. Finally a scenario of usage for this ontology in a knowledge-based system (KBS) has been shown.

The whole approach defines a brand new point of view on product variety. In literature it is usually seen as a source of complexity for manufacturing, planning and scheduling systems. Several works are focused on how to cope with this complexity. In this paper we dealt with the customer variety, representing the causes of the product variety and consequently of this complexity. Linking the product variety with the customer domain, we showed that a system for defining only the rationalised PL variety can be defined. On the manufacturing side, the aim is to delete the unjustified variety and therefore to reduce the unjustified complexity. On the customer side, the aim is to define configuration systems that do not require a product-related knowledge for configuring the right product.

It is important to highlight the consequence on the configuration systems discussed. On one hand these are completely customer-oriented: the customer is asked to describe its environment and what he expects from the product. On the other hand, all optimisations in configuration

system (e.g. price) are moved on the engineering side: a customer should represent a problem; only on the engineering side, appropriate algorithm can be deployed to identify the best solution (e.g. from a resources consumption point of view).

Future perspective of the present research concerns the development of the algorithms to finalise the structure of the KBS. A larger KB, even with knowledge from different domains, will be used to test the algorithm two. Moreover a more complex PL will be treated: according to the presented approach, the more complex the system-environment interactions are, the more complex is the implementation of the approach. Finally, an integration of this work with the proposal in (Giovannini et al., 2012) will be envisaged; the aim is to take into account the PL variety in an enterprise network.

9. REFERENCES

- Akao, Yoji (2004), *Quality Function Deployment: Integrating Customer Requirements Into Product Design*. Productivity Press.
- Aldanondo, M. and Vareilles, E. (2008), *Configuration for Mass Customization: How to Extend Product Configuration Towards Requirements and Process Configuration*. *Journal of Intelligent Manufacturing*, 19(5): 521–535.
- American Society of Heating, Refrigerating and Air-Conditioning Engineers, inc (2012), *2012 ASHRAE Handbook: Heating, Ventilating, and Air-conditioning Systems and Equipment*. Atlanta, Ga.: ASHRAE.
- Atkinson, Colin and Kiko, Kilian (2008), *A Detailed Comparison of UML and OWL*.
- Baader, Franz (2003), *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge university press.
- Balakrishnan, PV Sundar and Jacob, Varghese S. (1996), *Genetic Algorithms for Product Design*. *Management Science*, 42(8): 1105–1117.
- Von Bertalanffy, Ludwig (1956), *General Systems Theory*.
- Blecker, T., Abdelkafi, N., Kaluza, B. and Kreutler, G. (2004), *Mass Customization Vs. Complexity: A Gordian Knot?*, 890–903.
- Blockley, David I. and Godfrey, Patrick (2000), *Doing It Differently: Systems for Rethinking Construction*. Thomas Telford.
- Bouffaron, F., Gouyon, D., Dobre, D. and Morel, G. (2012), *Revisiting the Interoperation Relationships Between Systems Engineering Collaborative Processes*, in: *14th IFAC Symposium on Information Control Problems in Manufacturing, INCOM 2012*. Elsevier, IFAC Papersonline.
- Dassisti, Michele and Nicolò, Michele De (2012), *Enterprise Integration and Economical Crisis for Mass Craftmanship: A Case Study of an Italian Furniture Company*, 113–123, in: Herrero, P., Panetto, H., Meersman, R., and Dillon, T. (Eds.), *On the Move to Meaningful Internet Systems: OTM 2012 Workshops*. Lecture Notes in Computer Science. Springer.
- Davis, Stanley M. (1987), *Future Perfect*. Addison-Wesley.
- Du, X., Jiao, J. and Tseng, M. M. (2001), *Product Platform Representation: a Graph Grammar Approach*. *Manufacturing Systems*, 31(2): 1–7.
- Faisandier, Alain (2013), *Systems Architecture and Design*. Belberaud: Sinergy'Com.
- Ferryanto, Liem (2007), *DESIGN FOR SIX SIGMA*. *Jurnal Teknik Industri*, 9(1): 1-14.
- Gao, P., Lin, L. F., Cai, M. and Dong, J. X. (2003), *Automatic Acquisition of Product Configuration Model Based on Ontology Mapping*. *Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing System(China)*, 9(9): 810–816.
- Giovannini, A., Aubry, A., Panetto, H., Dassisti, M. and El Haouzi, H. (2012), *Ontology-Based System for Supporting Manufacturing Sustainability*. *Annual Reviews in Control*, 36(2): 309–317.
- Giovannini, A., Aubry, A., Panetto, H. and El Haouzi, H. (2013), *Mass Customisation in Sustainable Networked Enterprises*, 657–666, in: *Proceedings of the 14th IFIP Working Conference on VIRTUAL ENTERPRISE*, Sept. 30th-Oct. 2nd, Dresden, Germany, Collaborative Systems for Reindustrialization. Springer.
- Green, Paul E. and Srinivasan, Venkatachary (1978), *Conjoint Analysis in Consumer Research: Issues and Outlook*. *Journal of consumer research*: 103–123.
- Hazelrigg, G. A. (1998), *A Framework for Decision-Based Engineering Design*. *Journal of Mechanical Design*, 120(4): 653–658.
- Helo, P. T., Xu, Q. L., Kyllönen, S. J. and Jiao, R. J. (2010), *Integrated Vehicle Configuration System—Connecting the Domains of Mass Customization*. *Computers in Industry*, 61(1): 44–52.
- Hong, G., Hu, L., Xue, D., Tu, Y. L. and Xiong, Y. L. (2008), *Identification of the Optimal Product Configuration and Parameters Based on Individual Customer Requirements on Performance and Costs in One-of-a-kind Production*. *International Journal of Production Research*, 46(12): 3297–3326.
- Huffman, Cynthia and Kahn, Barbara E. (1998), *Variety for Sale: Mass Customization or Mass Confusion?* *Journal of retailing*, 74(4): 491–513.
- IEEE (2002), *IEEE Guide for CASE Tool Interconnections - Classification and Description*, 1175.1-2002.
- Inakoshi, H., Okamoto, S., Ohta, Y. and Yugami, N. (2001), *Effective Decision Support for Product Configuration by Using Cbr*, in: *Proceedings of the Fourth International Conference on Case-Based Reasoning (ICCBR), Workshop Casebased Reasoning in Electronic Commerce*, Vancouver, Canada.
- ISO/IEC (2007), *Systems and Software Engineering - Recommended Practice for Architectural Description of Software-Intensive Systems*. ISO/IEC 42010:2007. Geneva, Switzerland: International Organization for Standards (ISO)/International Electrotechnical Commission (IEC).
- ISO/IEC (2008), *Systems and Software Engineering - System Life Cycle Processes*. ISO/IEC 15288:2008 (E). Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC).

- ISO/IEC/IEEE (2007), Systems and Software Engineering - Measurement Process. ISO/IEC/IEEE 15939:2007. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC).
- ISO/IEC/IEEE (2011), Systems and Software Engineering - Requirements Engineering. ISO/IEC/IEEE 29148. Geneva, Switzerland: International Organization for Standardization (ISO)/International Electrotechnical Commission/ Institute of Electrical and Electronics Engineers (IEEE), (IEC).
- Jackson, Michael and Zave, Pamela (1995), Deriving Specifications from Requirements: An Example, 15–24, in: Proceedings of the 17th international conference on Software engineering.
- Janitza, Dennis, Lacher, Martin, Maurer, Maik, Pulm, Udo and Rudolf, Henning (2003), A Product Model for Mass – Customisation Products, 1023–1029, in: Palade, V., Howlett, R., and Jain, L. (Eds.), Knowledge-Based Intelligent Information and Engineering Systems. Lecture Notes in Computer Science. Springer Berlin / Heidelberg.
- Jiao, Jianxin (Roger), Simpson, Timothy W. and Siddique, Zahed (2007), Product Family Design and Platform-based Product Development: a State-of-the-art Review. *Journal of Intelligent Manufacturing*, 18(1): 5–29.
- Krishnan, V. and Gupta, Saurabh (2001), Appropriateness and Impact of Platform-Based Product Development. *Management Science*, 47(1): 52.
- Li, Bin, Chen, Liping, Huang, Zhengdong and Zhong, Yifang (2006), Product Configuration Optimization Using a Multiobjective Genetic Algorithm. *The International Journal of Advanced Manufacturing Technology*, 30(1): 20–29.
- Liker, Jeffrey K. and Morgan, James (2011), Lean Product Development as a System: A Case Study of Body and Stamping Development at Ford. *Engineering Management Journal*, 23(1): 16–28.
- Liu, X. B., Yuan, C. F., Gao, T. Y., Sun, W. and Wang, W. L. (2003), Layered Product Configuration Model Based on Feature and Faced to Customer. *Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing System(China)*, 9(7): 527–531.
- Mazo, R., Salinesi, C., Diaz, D., Djebbi, O. and Lora-Michiels, A. (2012), Constraints: The Heart of Domain and Application Engineering in the Product Lines Engineering Strategy. *International Journal of Information System Modeling and Design (IJISMD)*, 3(2): 33–68.
- Morgan, James M. and Liker, Jeffrey K. (2006), *The Toyota Product Development System*. Productivity press New York.
- Pohl, K., Bockle, G. and Van Der Linden, F. (2005), *Software Product Line Engineering*. Springer.
- Protégé available at <http://protege.stanford.edu/>
- Pyster, A., Olwell, D., Squires, A., Hutchison, N., Enck, S. and Anthony, J. (2012), *A Guide to the Systems Engineering Body of Knowledge (SEBoK)*. Version 1.0. Stevens Institute of Technology.
- Qin, Y. H. and Wei, G. X. (2010), Product Configuration Based on Modular Product Family Modelling. *Journal of Computational Information Systems*, 6(7): 2321–2331.
- Shah, Rachna and Ward, Peter T. (2007), Defining and Developing Measures of Lean Production. *Journal of operations management*, 25(4): 785–805.
- Sommerville, Ian and Sawyer, Pete (1997), *Requirements Engineering: A Good Practice Guide*. New York, NY, USA: John Wiley & Sons, Inc.
- Spear, Steven and Bowen, H. Kent (1999), Decoding the DNA of the Toyota Production System. *Harvard Business Review*, 77: 96–108.
- Tseng, Mitchell M. and Du, Xuehong (1998), Design by Customers for Mass Customization Products. *CIRP Annals - Manufacturing Technology*, 47(1): 103–106.
- Wang, YAPING, Han, GUIHUA, Ge, JIANGHUA, Qi, JINGRUI and Xu, JIANYUAN (2011), Research on Design Method of Demand-driven Product Configuration for Mass Customization. *Journal of Advanced Manufacturing Systems*, 10(1): 117–125.
- Wassenaar, Henk Jan and Chen, Wei (2001), An Approach to Decision-based Design, , in: Proceedings ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Pittsburgh, Pennsylvania, DETC2001/DTM-21683.
- Xie, H., Henderson, P. and Kernahan, M. (2005), Modelling and Solving Engineering Product Configuration Problems by Constraint Satisfaction. *International Journal of Production Research*, 43(20): 4455–4469.
- Yang, Dong and Dong, Ming (2012), A Constraint Satisfaction Approach to Resolving Product Configuration Conflicts. *Advanced Engineering Informatics*, 26(3): 592–602.
- Zhou, Chunjing, Lin, Zhihang and Liu, Chuntao (2008), Customer-driven Product Configuration Optimization for Assemble-to-order Manufacturing Enterprises. *The International Journal of Advanced Manufacturing Technology*, 38(1): 185–194.
- Zhu, Bin, Wang, Zhan, Yang, Haicheng, Mo, Rong and Zhao, Yanwei (2008), Applying Fuzzy Multiple Attributes Decision Making for Product Configuration. *Journal of Intelligent Manufacturing*, 19(5): 591–598.