Chapter 14
Modularity and Supporting Tools and Methods

Josip Stjepandić, Egon Ostrosi, Alain-Jérôme Fougères and Martin Kurth

Abstract The paradigm of modularity has emerged as a relevant way to meet customer requirements with a wide range of variety and customisation of products, from unique to standard ones. The modularity area is becoming increasingly multidisciplinary, which implies holistic and articulated concurrent engineering approaches. Modularity can intersect technical aspects with the business aspects. The use of modular technology has wide-reaching implications for any design and development company that undertake to use this paradigm. This chapter provides a framework for understanding the modularity in the context of concurrent engineering. It involves design for modularity as well as management of modularity. Theoretical and practical development of consistent modular methods, their implementation technologies and tools for mass customization and product configuration are examined. Some of the possible implications of these developments are presented from concurrent engineering point of view. The current trend is drawn toward usage and integration of different technologies such as advanced CAD systems, product configurators, agent-based systems and PDM systems. Three particular application areas with industrial use cases are presented. A discussion about research challenges and further developments closes this chapter.

Keywords Modularity · Modular design · Product variety · Mass customisation · Product platform · Product configurator

J. Stjepandić (✉)
PROSTEP AG, Dolivostr. 11, 64293 Darmstadt, Germany
e-mail: josip.stjepandic@prostep.com

E. Ostrosi · A.-J. Fougères
Université de Technologie de Belfort-Montbeliard, Belfort 90010, France
e-mail: egon.ostrosi@utbm.fr

A.-J. Fougères
e-mail: alain-jeromefougeres@utbm.fr

M. Kurth
RAYCE EURL, Lörrach 79539, Germany
e-mail: martin.kurth@rayce-lab.com

J. Stjepandić et al. (eds.), *Concurrent Engineering in the 21st Century*,
DOI 10.1007/978-3-319-13776-6_14

14.1 Introduction

Many sectors of human existence are tightly connected with the term of modularity. Besides techniques, modularity is intensively used in education, science (mathematics, informatics, psychology, biology and linguistics), media science, management, organization, financial services and the public administration. Modular products accompany practically a person throughout his life. As a commonly known artifact, the entire Web has a modular structure, composed of independent sites and pages, and each webpage itself is composed of elements and code that can be independently modified and can interact via clearly defined interfaces [1].

Modularity as an engineering and management domain has become relevant in the 60 s of the twentieth century through the design of the first modular computers. Through the development of concepts and a body of knowledge modularity has become an area worthy of study in its own right. It can be considered that the roots of modularity can be derived from human cognitive abilities [2]. In the 1980s Fodor revived the idea of the modularity of mind, although without the notion of precise physical localizability. According to Fodor modular (cognitive) systems fulfill certain criteria:

1. Domain specificity: modules only operate on certain kinds of inputs
2. Informational encapsulation: modules need not refer to other psychological systems in order to operate
3. Obligatory firing: modules process in a mandatory manner
4. Fast speed: probably because modules are encapsulated and mandatory
5. Shallow outputs: the output of modules is very simple
6. Limited accessibility
7. Characteristic ontogeny: there is a regularity of development
8. Fixed neural architecture.

These criteria are also valid in equal measures for modular technical systems. The precise definition of product modularity is provided by articulating a product system modularity construct in the domain of tangible assembled artifacts [3]. It focuses product modularity to the criteria of component separability and component combinability. This definition is finally related to other definitional perspectives synthesized by a literature review: component commonality, function binding, interface standardization, and loose coupling. The nomological network of the product modularity construct is derived from it and subject to further validation.

In context of concurrent engineering, modularity combines technical aspects with business aspects, both from a qualitative and a quantitative viewpoint (Fig. 14.1).

Complex products can be understood as a network of components that share technical interfaces (or connections) in order to function as a whole. Component modularity is defined based on the lack of connectivity between components [4]. Technically (Fig. 14.1), it can be expressed with three measures: (a) how components share direct interfaces with adjacent components, (b) how design interfaces

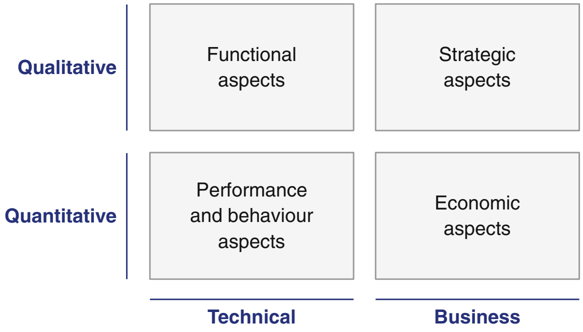


Fig. 14.1 Aspects and viewpoints of modularity

may propagate to nonadjacent components in the product, and (c) how components may act as bridges among other components through their interfaces. All three measures of component modularity have been identified for the product architecture of a large commercial aircraft engine. While trying to redesign components it was detected that the relationship between component modularity and component redesign depends on the type of interfaces connecting components.

Component commonality—the use of the same version of a component across multiple products—is being increasingly considered as a promising way to offer high external variety while retaining low internal variety in operations. As components influence to a greater or lesser extent nearly every process step along the supply chain, a multitude of diverging commonality problems is being investigated in literature. Representation of networks of components by graphs and the development of graph-based approaches have been applied as flexible and efficient means to a wide range of commonality problems [5].

Product-family design and platform-based product development also use the concept of modularity. Decision frameworks have been also introduced in this context to reveal a holistic view, encompassing both front-end and back-end issues such as product portfolio and product family positioning, platform-based product family design, manufacturing and production, and finally supply chain management [6].

From the business point of view (Fig. 14.1), modularization has three purposes: to make complexity manageable, to enable parallel work, and to accommodate future uncertainty [7]. The impact of modularity to the financial and organizational structure of an industry can be described with three aspects: (1) Modularity is a financial force that can change the structure of an industry; (2) The value and costs associated with constructing and exploiting a modular design are explored; (3) The ways in which modularity shapes organizations and the risks that it poses for particular enterprises are examined.

Modularization in enterprise leads, thus, to the disaggregation of the traditional form of hierarchical governance. The enterprise is decomposed into relatively small autonomous organizational units (modules) to reduce complexity and to integrate strongly interdependent tasks while the interdependencies between the modules are weak. The dissemination of modular organizational forms yields a strong process orientation: the complete service-provision process of the business is split up into partial processes, which can then be handled autonomously by cross-functional teams within subunits [8].

This chapter examines the main developments and implementation of modularity in the context of CE. In Sect. 14.2, the foundations of modularity are highlighted from design and management perspective. Approaches for support of modular design are explained and compared in Sect. 14.3. Subsequently, technologies and tools for modular design are introduced in Sect. 14.4. Three industrial use cases (automotive, aerospace, plant design) are described in Sect. 14.5. A discussion chapter gives insight into further development in field of modularity. Finally, an outlook is given with respect to the future of modularity from a CE perspective.

14.2 Modularity: Design and Management

In general, from a management perspective modularity is seen as a business strategy for efficient structuring of complex products, procedures and services with the objective to rationalize the enterprise [9]. A modular system (product, procedure, service) is, in contrast to monolithic systems, composed of separate modules, which satisfy Fodor’s criteria and can be used in different product variants. It is possible to develop modules independently of each other and then bring them together in an integrated whole, a product, a procedure or a service on the market. Basically, a module (as a building block or black box) is interchangeable with another one. By fusing of different modules the range of products and services (solution space, range of articles) is enlarged. Opposite to an ad hoc reuse, this approach invests selectively in systematic reuse ability for later benefit.

14.2.1 Modular Design

Modular design combines both, the benefits of standardization and the benefits of customization. The advantages of modularity result from the reuse of parts and the repetition of operations. Hence, higher quality, lower cost of production, lower delivery time and easier spare parts procurement are expected [10]. Modularity facilitates collaboration and thereby helps to increase flexibility and to minimize economic risk. Since many modules already exist and can be purchased from an external vendor, the best module available (outsourcing) can be selected. Because of mutual independency a module is easier to design, produce, test, maintain and repair than a single more complex system [11]. As a result, a large product variance can be covered with a minimum quantity of modules.

The disadvantages of modular products are a limited selection within the supply range compared to a (entirely) customized product or service as well as high sensitivity to possible design errors in a single module or, even worse, in the interface between two modules. Furthermore, the function of each product variant can never be optimal, while some variants may be over-dimensioned. Consequently, product differentiation for the customer may be inhibited or lost [12].

Modular design considers functions, properties and interfaces of product constituents. Standard interfaces make parts interchangeable, thereby reducing the expenditure for the combination of different product constituents. Modular design usually involves the following processes: the identification of product architecture and reusable components (building blocks) from existing products, the agglomeration and adaptation of singular building blocks into modules to derive a new design, and assessment of product performance and cost. Modular product architecture is generated by deriving a rule base (scheme) for the mapping of product functions to physical components. For the utilization of modules comprehensive interfaces become crucial. A modular architecture is distinguished from an integral architecture by the way functional elements are mapped to physical components: it has a one-to-one mapping (design logic) from functional elements to physical components of the product. Three basic types of modular architecture are defined, namely slot, bus and sectional, according to the interfaces between components [13].

Platforms as a special expression of a modular design are of particular relevance for an industrial practice. A platform is a standardized base product with fundamental functions and properties of the total product, on which a variety of similar products can be efficiently built by using subsystems, modules and components. In the platform the architecture and the interfaces to optional elements are included, which are used for differentiation of the end products [14].

14.2.2 Mass Customization, Variety and Configuration

Under the term “Mass Customization” a business strategy is defined that utilizes modular design for complex offerings of products and services that are configured on demand to achieve the best fit with customer-specific needs [15]. Product variety and customization are provided through flexibility and quick responsiveness (“Configure-to-Order”). Mass customization joins two concepts that are usually supposed to be opposite: mass production and customization including two approaches: mass and craft (single-piece) production. Mass production manufactures low cost products by reaping the benefits of standardization and scale economies. On the other hand, craft production assumes a high level of individualization since the products are tailored to specific customer requirements.

Opposite to entirely individual products from special-purpose production customization options are allowed for allocated product structure areas only. Similarly, the processes of order processing are highly standardized, in order to quickly and cost-efficiently fulfill individual customer requirements. One of the reasons to establish mass customization is the replacement of the strategy build-to-stock (BTS) which refers to products that are built before a final purchaser has been identified, with production volume driven by historical demand information. BTS becomes inflexible with new market demands. To prevent costs expansion by large finished goods inventory, an introduction of a build-to-order (BTO) strategy is necessary to become a mass customization manufacturing enterprise [16]. By moving the customer order decoupling point (CODP) upstream in the value creating chain, the customer influence is increased and the response time shortened. In case of the automotive industry it realizes the delivery to the customer of a bespoke vehicle 5 days after placing the order [17].

Product structures of customized products must be thoroughly adjusted for specific customization options by adopting entirely individual components that are specifically created besides of standardized and configurable modules. Generally, a fixed and a variable area of product structure can be identified, in which mandatory and optional spaces are foreseen for individual implementation. In such a structure, technologies and tools as follow in Sect. 14.4 are useful, supported by CE technologies like Knowledge-based Engineering (see Chap. 10) and Digital Mock-up (see Chap. 13). Management of such a flexible, change-robust product structure is predetermined in a superordinate PLM concept (see Chap. 16).

Mass customization heavily affects all phases in the product creation process. Product customization is usually supported by configuration systems (see Sect. 14.4.2). Generic conceptual procedures for designing such systems are important for mass customization. These procedures involve analysis and redesign of the business processes, which can be supported by a configuration system, analysis and modeling of the company's product portfolio, selection of configuration software, programming of the software, and implementation and further development of the configuration system [18].

14.2.3 Modularity from a Management Perspective

By now, modularity has become a basic irreplaceable development methodology inside the product strategy for a variety of technical products planning based on market research and correspondent forecast. Individual products are not developed anymore, only whole product families or product spectra. A later change on this strategy would be very expensive and could implicate massive negative effects [19]. Development of new modules is essentially a new innovation process.

Modularity seems counter-productive, when selective distinctive features are the reason to buy a product. When customers focus on elements, like styling, haptics, or specific colours creative freedom is necessary. In such cases modular design is not applicable, because investments in modular design outweigh the efforts to create a user-specific product of which the number is often very small.

With the introduction of modular design, project control has to be adapted when focused on individual projects. The funding of the development of individual components has to be increased for every development project for a modular product family. The integration of different product variants does not come with any monetary benefits if it is not organized through a holistic controlling approach [20].

This approach enables the assessment of modular product families as well as their holistic management based on the new modularity-balanced-score-card (M-BSC). Additionally, the different perspectives from production, development, marketing and sales need to be integrated.

Cost schemes of modular products can be established by decomposing the product family into generic modules to support cost calculation [21]. The candidate modules that are closest to the customer requirements can be retrieved by computing a similar degree of the case module and the target module for the same module model quantitatively. The search space is restricted for generic modules. The cost structure of the different types of modules is analyzed, i.e., the basic module and customized module, and the different cost estimation approaches are applied to different types of modules. The product cost can be accumulated with cost of modules in each level of the modular product family progressively.

14.3 Methodical Support for Modular Design

Modularity is achieved by partitioning information into visible design rules and hidden design parameters. It is beneficial only if the partition is precise, unambiguous, and complete [9].

The visible design rules (“visible information”) are basic decisions that affect subsequent design decisions. Ideally, the visible design rules are established early in a design process and communicated broadly to those involved. Visible design rules consist of three categories:

1. An architecture, which specifies system modules and their functions.
2. Interfaces with description of module interaction (fit, connect, communicate).
3. Standards for testing a module” conformity to the design rules and for comparing performance of competing modules.

The hidden design parameters (“hidden information”) are decisions that affect the design only within the local module. Hidden parameters can be used late and changed whenever necessary.

Common attributes of modular products are defined as follows [16]:

1. *Commonality of modules*: Components or modules are used at various positions within a product family.
2. *Combinability of modules*: Products can be configured by combining components or modules.
3. *Function binding*: There is a fixed allocation of functions to modules.
4. *Interface standardization*: The interfaces between the modules are standardized.
5. *Loose coupling of components*: The interactions between the components within a module are significantly higher than the interactions between components of various modules.

In modularization the combinability of elements is increased when starting from an existing production structure [22]. Thereby, through reduction of interdependence a larger amount of product variants can be generated from a given number of element variants. Another opportunity concerns reduction of the level of integration. This can be achieved by interface standardization.

There are various methods to support modular design like axiomatic design (AD), functional modeling, design structure matrix (DSM), modular function deployment (MFD) and variant mode and effects analysis (VMEA), which can be also used in combination with an architecture development process [23]. The interdependence and the level of integration of components are a measure of modularity. Classification of various product modularization methods and the comparison of methods in several application areas (product variety, product generation and product lifecycle) have shown that the generation of modules depends on both the chosen method and the weighting of different criteria [24].

14.3.1 Axiomatic Design

Axiomatic design (AD) is a systems design methodology using matrix methods to systematically analyze the transformation of customer needs into functional requirements, design parameters, and process variables [25]. Hereby, the attempt is made to build on the development of new products based on a system of axioms, which are based on mathematics or physics sciences. The formalization is supposed to lead to the design of technical systems. Starting from two axioms (the independence axiom and the information axiom) a system of theorems is set up.

AD contains four domains: customer domain, functional domain, physical domain and process domain. Each domain has its corresponding design elements, namely, customer attributes (CAs), functional requirements (FRs), design parameters (DPs), and process variables (PVs). The zigzag mapping among domains follows the process of product design. The independence axiom, which demands maximizing the independence of the functional requirements, is used to judge the rationality of design. The information axiom, which demands minimizing the information contents of the design, is used to select the optimum design.

The mapping process can be expressed mathematically in terms of the characteristic vectors for functional requirements (FRs) and design parameters (DPs) that define the design goals and design solutions. The relationship between the two vectors can be expressed by the design equation for product design as following:

$$\{\mathrm{FRs}\} = [\mathrm{A}]\{\mathrm{DPs}\}, \tag{1}$$

where [A] is a matrix defined as the design matrix that characterizes the product design. Ideally, it's a square matrix (when the number of FRs is equal to the number of DPs). When the design matrix is either diagonal or triangular, the corresponding design satisfies the independence axiom ("uncoupled design" resp. "decoupled design"). Any other form of design matrix is called full matrix and will result in a coupled design that is not beneficial. The uncoupled design indicates that the design tasks are mutually independent, and can run concurrently shortening the development time significantly. The decoupled design indicates that the design tasks should be processed by sequence so that the whole process can be managed effectively.

The modular design based on AD runs in three major steps:

1. *Analysis of product using AD*: The fundamental functions requirements and design parameters are decomposed into levels of sub-function requirements and sub-design parameters resulting the functional-structure model and a full design matrix.
2. *Module definition*: Product design is implemented agglomerating the modules from the singular functions and the components in the functional-structure model.
3. *Reconfiguring the sequence of modules*: The relationships between modules are uncoupled or decoupled on the basis of AD. The uncoupled relationships mean that these modules can be carried out simultaneously and the decoupled relationships mean that these modules must be performed in sequence so that the effect of former modules can be considered and the iterations of design can be reduced. In this step, we can get the module-junction structure diagram that indicates the design sequence.

Mapping of the modular product architecture follows the axiomatic design to maintain the cross-domain module independence of functional requirements. Aerospace and defense systems have been analyzed as complex systems with long lifecycles with particular attention to time and flexibility [26]. By dividing the design perspective into two domains, one long-term architectural and one short-term modular, and identifying an ideal product architecture for that situation, the design in terms of integral architecture and modular flexibility was optimized and the resolution of the issue with conflicting short and long term goals supported.

14.3.2 Functional Modeling

Functional modeling place product functions as a central idea for structuring of product into modules. The role of functional modeling is not only to clarify understanding of a design problem, but also to serve as core to the modular solution. Questions arise from these roles of functional modeling. *How should functions can be expressed or represented so that their structure can be used for modular design? What are rules or heuristics to identify modules from patterns in function structure?*

Functional modeling is based on the assumption that product functions can be decomposed into smaller, easily solvable sub-functions. The modeling of functions as operations on flows allows designers to represent product functions and to decompose these functions into chains of connected elementary sub-functions. The sub-functions in these chains are related by the flow of energy, material or signal passing through the product to form a functional model, known as a function structure. The product functions and sub-functions are described in a verb-object form. They are represented by a black-boxed operation on flows of energies, materials, and signals.

Modularity and the modular form of the product depend on the form of chains of connected elementary sub-functions. The modular design based on the function decomposition follows these steps [28]:

1. *Define product functions*. Product functions are defined as operations on flows of energies, materials, and signals. They originate from customer needs.
2. *Decompose product functions into sub-functions*. For each product function, decompose the flows of energies, materials, and signals. For each input flow, define a chain of sub-functions that transforms the flow step-by-step into an output flow. Because sub-functions are modeled also as operations on flows, they can be standardized. A library of flows and sub-functions has been proposed by Hirtz et al. [27]. These libraries are called functional bases. A designer can decompose product functions into sub-functions chosen from the library of sub-functions (Table 14.1). Modeling of sub-functions as operations on flows implies a temporal order relationship between sub-functions.
3. *Integrate the chains of sub-functions*. The temporally ordered chains of sub-functions are integrated by connecting the chains. Thus, series of sub-functions, sequentially and/or in parallel, transforms input flows step-by-step into output flows. The decomposed product functions form now a function structure.
4. *Identify modules*. Groups of sub-functions related by flows can be observed in the function structure of the product to form modules. This observation has led to the formulation of heuristics to identify modules. Stone and Wood [28] proposed three heuristics based on the three patterns that a flow can experience: 1) a flow may pass through a product unchanged, 2) a flow may branch, forming independent function chains, or 3) a flow may be converted to another type. Figure 14.2 shows the overall function of an electric wok with its function structure with heuristically identified modules [28].

This method provides a systematic technique for identifying modular patterns at the functional model stage. This method also allows product architecture decisions to begin at a much earlier stage, i.e. at the functional model stage. Once modular patterns in functional structure are defined, the design of alternative layouts and components become more straight forward tasks. Optimization techniques may then be applied to find optimal modular product.

14.3.3 Design Structure Matrix

The design-structure-matrix (DSM) is a simple and meanwhile commonly used tool for the visualization of complex connections in systems in a compact and easily adjustable format. This facilitates their analysis. The systems can be both technical and social (e.g., organizations) [29].

Table 14.1 Library of sub-functions [27]

Primary	Secondary	Tertiary
Branch	Separate	Divide
		Extract
		Remove
	Distribute	
Channel	Import	
	Export	
	Transfer	Transport
		Transmit
	Guide	Translate
		Rotate
		Allow DOF
Connect	Couple	Join
		Link
	Mix	
Control	Actuate	
	Regulate	Increase
		Decrease
Magnitude	Change	Increment
		Decrement
		Shape
		Condition
	Stop	Prevent
		Inhibit
Convert	Convert	
Provision	Store	Contain
		Collect
	Supply	
Signal	Sense	Detect
		Measure
	Indicate	Track
		Display
	Process	
Support	Stabilize	
	Secure	
	Position	

The DSM is represented as a square N × N matrix, mapping the interactions among the set of N system elements. Depending on the type of system being modeled, DSM can represent various types of architectures (product, process, organization, multiple domains). For example, to model a product” architecture, the DSM elements would be the components of the product, and the interactions would

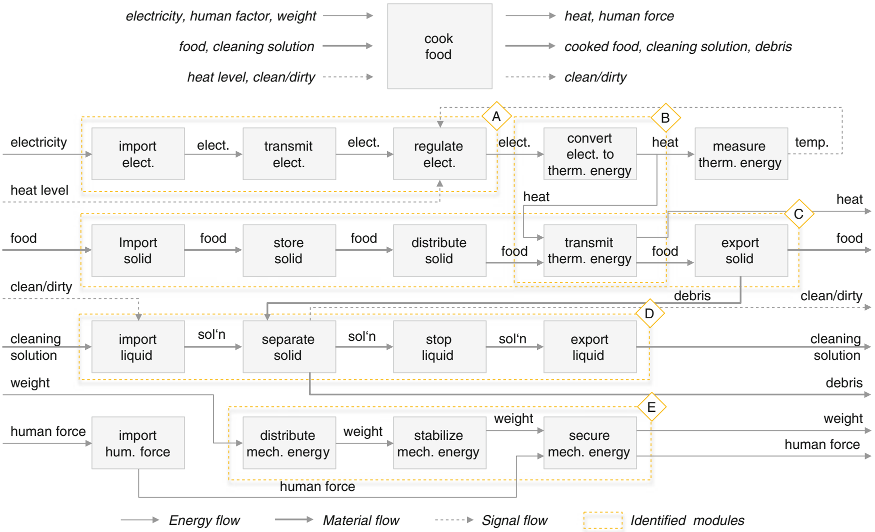
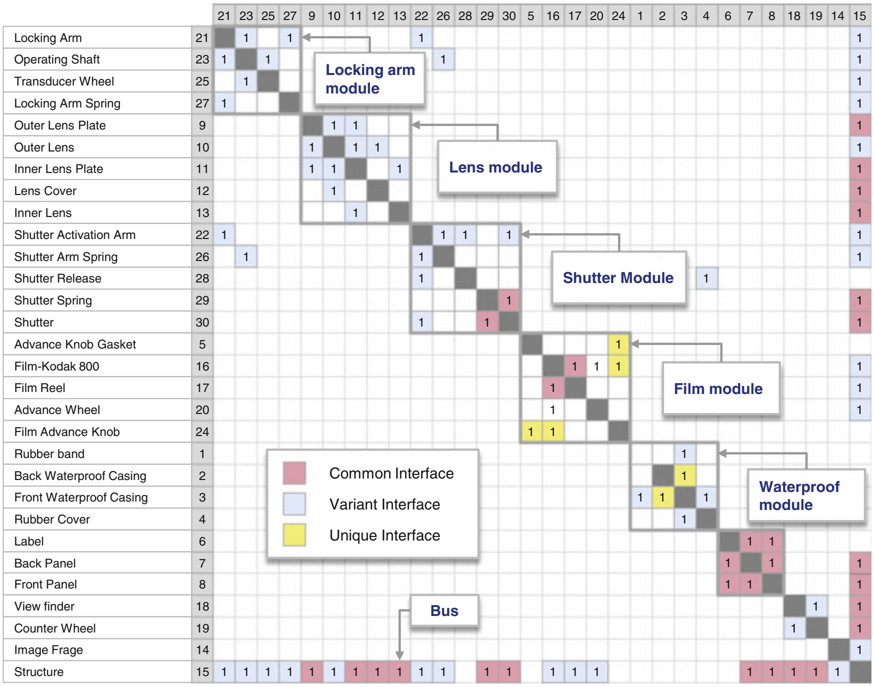


Fig. 14.2 An overall function of an electric wok with its function structure with heuristically identified modules [28]

be the interfaces between the components. Using the advanced three-dimensional DSM (DSM^{3D}), designers are able to highlight the differences among singular members within families of products, modules, and interfaces.

The DSM can be illustrated by the modular design of a single-use camera [29, p. 75] (Fig. 14.3). Through marks outside the diagonal (21, 21–15, 15) it is shown, that one component in the system is dependent on another one. Reading across a row reveals what to what other elements the element in that row provides outputs, and scanning a column reveals from what other elements the element in that column receives inputs. In the example, element 11 (inner lens plate) provides input for the elements 9, 10, and 13, simultaneously. Due to the symmetry, interdependent elements can be recognized: 9 needs 11 at an early stage and at the same time 11 is providing output to 9.

In this use case five Kodak single-use cameras were dissected. By comparing component interactions across the five camera models, interfaces were categorized as common (occurring in all five), variant (occurring in some of the five), or unique (occurring in one of the five) and colored (red, blue, yellow, respectively) into the matrix in Fig. 14.3. Based on interaction among components and appropriate clustering, five modules were identified indicated by square borders in the DSM. The last component in the list (structure) is related to many other components, as indicated with the many colored squares along the bottom of the DSM. This bus-type component is strategic because there is an opportunity to use common interfaces to save costs and better handle diversity.



		21	23	25	27	9	10	11	12	13	22	26	28	29	30	5	16	17	20	24	1	2	3	4	6	7	8	18	19	14	15
Locking Arm	21	■	1		1						1																				1
Operating Shaft	23	1	■	1								1																			1
Transducer Wheel	25		1	■																											1
Locking Arm Spring	27	1			■																										1
Outer Lens Plate	9					■	1	1																							1
Outer Lens	10					1	■	1	1																						1
Inner Lens Plate	11					1	1	■		1																					1
Lens Cover	12						1		■																						1
Inner Lens	13							1		■																					1
Shutter Activation Arm	22	1									■	1	1		1																1
Shutter Arm Spring	26		1								1	■																			1
Shutter Release	28										1		■											1							
Shutter Spring	29													■	1																1
Shutter	30										1			1	■																1
Advance Knob Gasket	5															■				1											
Film-Kodak 800	16																■	1	1	1											1
Film Reel	17																1	■													1
Advance Wheel	20																1		■												1
Film Advance Knob	24															1	1			■											
Rubber band	1																						1								
Back Waterproof Casing	2																					■	1								
Front Waterproof Casing	3																				1	1	■	1							
Rubber Cover	4																						1	■							
Label	6																								■	1	1				
Back Panel	7																								1	■	1				1
Front Panel	8																								1	1	■				1
View finder	18																											■	1		1
Counter Wheel	19																											1	■		1
Image Frage	14																													■	1
Structure	15	1	1	1	1	1	1	1	1	1	1	1		1	1		1	1	1							1	1	1	1	1	■

Fig. 14.3 Clustered DSM for modular design of a camera [18]

The DSM enables designers to study basic aspects of product architecture such as bus, mini-bus, and strength of physical interactions. It also helps to investigate the architectural distribution of modules and interfaces.

The DSM shows that many interfaces are variants. Interfaces and modules having instantiation in multiple products are named respectively cross-interfaces and cross-modules. By identifying cross-interfaces and cross-modules, which are beneficial from a cost saving point of view, designers should avoid variant interfaces (colored blue) that provide diversity and incur additional cost. In the case of variant cross-modules, the DSM indicates that it is necessary to develop a common cross module with common interfaces.

A small number of computer software applications incorporate dependency structure matrices in particular for software development. The DSM knowledge is maintained by the DSM Community, an interdisciplinary expert group [30].

A four-step product module identification approach by combining AD and DSM is proposed in [31] (Fig. 14.4). The overall pertinence DSM of DPs is obtained in first step, which assembles the three aspects of pertinence relationship of functionality, structure, and manufacturing process. The similarity DSM of DPs from manufacturing process is built in the second step. Based on the two steps above, the overall pertinence DSM with similarity DSM is aggregated and the overall

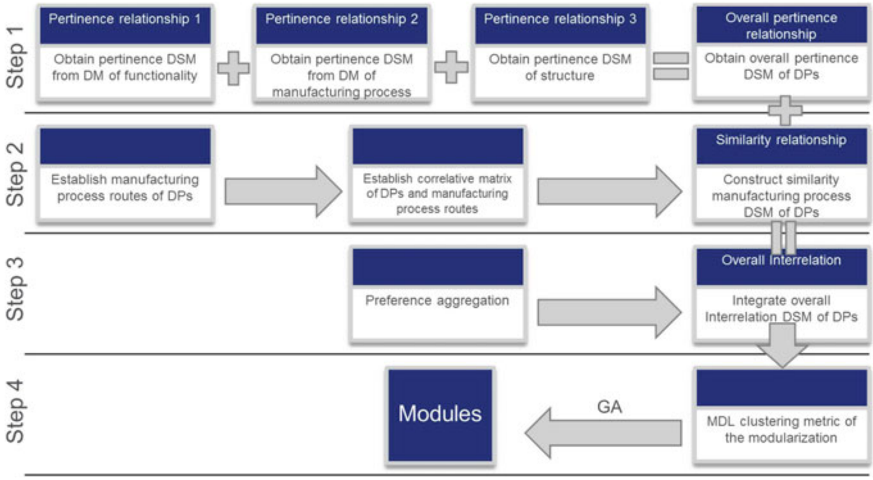

Step 1
Pertinence relationship 1
Obtain pertinence DSM from DM of functionality
Pertinence relationship 2
Obtain pertinence DSM from DM of manufacturing process
Pertinence relationship 3
Obtain pertinence DSM of structure
Overall pertinence relationship
Obtain overall pertinence DSM of DPs
Step 2
Establish manufacturing process routes of DPs
Establish correlative matrix of DPs and manufacturing process routes
Similarity relationship
Construct similarity manufacturing process DSM of DPs
Step 3
Preference aggregation
Overall Interrelation
Integrate overall Interrelation DSM of DPs
Step 4
Modules
GA
MDL clustering metric of the modularization

Fig. 14.4 Product module identification based on AD and DSM [31]

interrelation DSM of DPs is obtained in the third step. The overall interrelation DSM is clustered and modules generated by using genetic algorithms and minimal description length in the fourth step.

14.3.4 Modular Function Deployment (MFD)

As a further occurrence of a matrix method, the modular function deployment (MFD) approach was developed by Erixon [32] with the target to facilitate the development of a “robust” production program that is easily adaptable to the varying requirements. It is based on the quality function deployment (QFD) methodology, which is expanded with the modularity concept. The MFD approach introduces dedicated criteria (“module drivers”), which compile a business strategy into a framework for modular product design. Module drivers yield the basis for a systematic evaluation of technical solutions for a given product, based on an accommodated product structure.

MFD consists of five major steps (Fig. 14.5):

1. *Clarify customer requirements*: This step ensures that the precise design specification as functional requirements is derived from customer requirements e.g. by using QFD.
2. *Evaluation of functions and selection of technical solutions*: This step is also called “functional decomposition” because functions and sub-functions are identified and assigned to the technical solutions (“function drivers”) to fulfill the functional requirements. For each requirement there are many possible technical solutions.

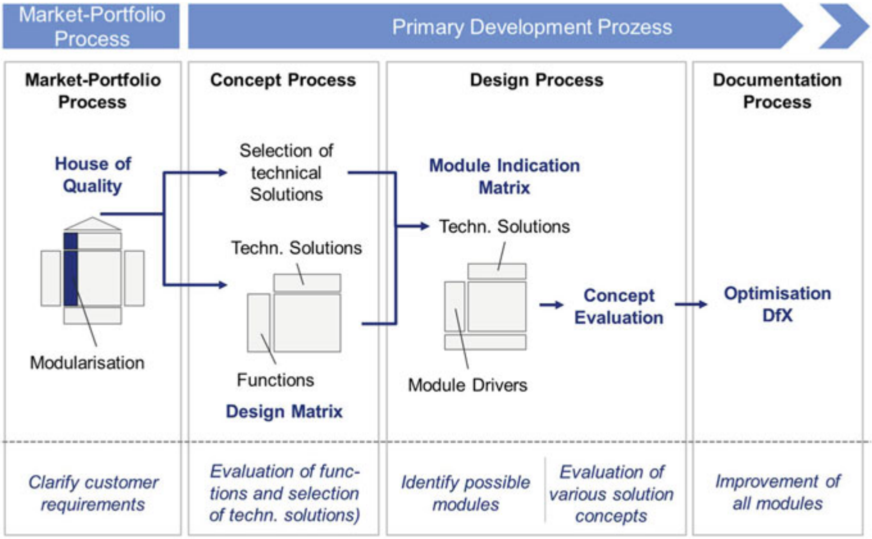


Fig. 14.5 Modular function deployment

3. *Identify possible modules*: The most important step contains the systematic generation and selection of modular concepts, in which the module indication matrix (MIM) is used to identify possible modules by examining the interrelationships between dedicated criteria ("module drivers") and technical solutions. A questionnaire is provided to support this activity.
4. *Evaluation of various solution concepts*: MIM also provides a procedure for investigating opportunities of integrating multiple functions into single modules by using the interface matrix. Herein, modules are first sorted in the expected assembly sequence. In this matrix is recorded which module are coupled by which kind of interface (so called "hamburger" or "base part assembly"). The expected effects of the redesign can be estimated and an evaluation can be provided for each modular concept.
5. *Improvement of modules*: The singular modules are improved—independent on each other by using appropriate ranking methods (e.g. Rank Order Clustering).

The overall understanding of MFD is that improved product modularity facilitates the entire flow of information and materials—from development and purchasing to logistics and delivery. The focus lies on the combination of function assignment and economic criteria ("module drivers"). It is a modularity shaping method, which integrates different weighed goals and produces a modular product architecture. Thus, multidimensionality is its special characteristic.

14.3.5 Variant Mode and Effects Analysis (VMEA)

The Variant Mode and Effects Analysis approach (VMEA) helps to depict the impacts of product variants in all units of the enterprise from definition of the product program to distribution. It includes an evaluation of target costs and discovers cost-saving potential by eliminating product variety that is not customer-perceived.

The VMEA is divided into three different levels; (1) basic VMEA, in the early design phase, when we only have vague knowledge about variation; the goal is to compare different design concepts, (2) advanced VMEA, further in the design process, when we can better judge the sources of variation, and (3) probabilistic VMEA, in the later design stages, where we have more detailed information about the structure and the sources of variation; the goal is assessment of reliability [33].

Advanced VMEA is used to optimize modularity [34]. A systematic development of variety diversity is supported according to technical and economic criteria. Optimal product structures are determined by variation of parts and modules. The VMEA integrates business units like marketing, product program planning, product development, production and distribution and is executed through the following steps (Fig. 14.6):

1. Market-oriented evaluation and design of product functions and determination of target costs.
2. Derivation of design alternatives for the realization of a homogenous product structure.
3. Evaluation of design alternatives for technical and economic aspects and selection of variant solution.
4. Definition of the product program and its transfer to sales.

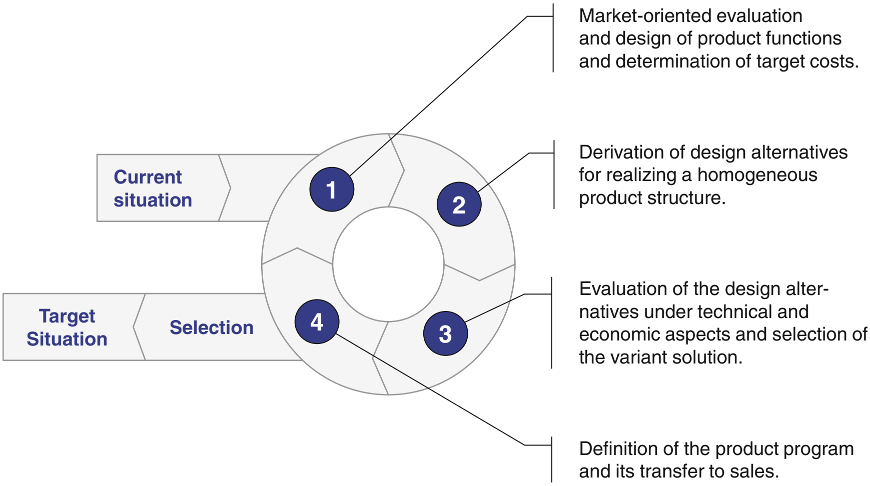

Current situation
Target Situation
Selection
1
2
3
4
Market-oriented evaluation and design of product functions and determination of target costs.
Derivation of design alternatives for realizing a homogeneous product structure.
Evaluation of the design alternatives under technical and economic aspects and selection of the variant solution.
Definition of the product program and its transfer to sales.

Fig. 14.6 Variant Mode and Effects Analysis (VMEA)

The primary benefit of VMEA is that the systematic preparation of multiplicity information and the graphic presentation of the development of variants can support the required communication between the units involved in VMEA: marketing/sales, development/design, work preparation/production and budgeting/controlling. Thus, the diversity of variants is known for future products and unneeded reserves concerning performance and flexibility of the equipment can be avoided.

14.4 Technologies and Tools for Modular Design

Currently, manifold technologies and tools are offered to foster modular design:

- Select pre-defined components and assemblies (standard parts library/catalog),
- Configure the product based on customer demand (product configurator),
- Facilitate the saving and selecting of knowledge of modular products (agent-based approach) or
- Facilitate definition and maintenance of general product structures, so called "150 % product structure" (PDM system).

These technologies have been developed independently and support specific aspects of modular design. They can provide optimal functionality by mutual integration and interaction with other systems (e.g. ERP). If their mutual customization is conducted thoroughly, all characteristics of modular design can be mapped.

14.4.1 Standard Parts Catalog/Library

The module standard parts catalog, which exists in every modern CAD system, is an easy way to foster product modularity, not only because it provides standard parts (according to national and international standards), but also because it facilitates the insertion of external part libraries and carry-over parts as a CAD functionality. In this case, the designer's activity is limited to the selection, insertion and validation of such parts in an assembly.

Based on this basic CAD functionality, enterprise part library solutions have been developed, which facilitate the allocation, administration and integration of external part libraries (e.g., which are provided by component suppliers) as an alternative for individual (printed or digital) catalogues. Thereby, extent and selection of purchased parts can be limited and prioritized according to individual criteria by a central standardization department in cooperation with the purchase department (e.g., parts of component suppliers with the best quality certificate are preferred). This will lead to massive cost savings. As a result of such search a designer gets 3D and/or 2D models in the desired native or neutral format (STEP, IGES, DXF) plus metadata for down-stream applications.

The administrator has the possibility to maintain the database in real-time by considering the business activities of a supplier (e.g., extension of its product portfolio). Also, regional differences in delivery can be considered. In case a standard part is required, which is not available, a pre-defined work flow exists to automate the generation of requests and to provide tracing, justifying and monitoring the processing of requests, and for reporting nonconformities. In addition to a part library a search engine for 3D data can be used that locates similar parts in large, heterogeneous data resources within fractions of a second. It analyzes the 3D geometry of the parts and automatically extracts characteristic features and facilitates part reuse in such way.

14.4.2 Product Configurator

A product configurator is a multi-functional, commercial IT tool which serves as interface between sales and delivery in an enterprise. It supports the product configuration process so that all design and configuration rules, as expressed in a product configuration model, are guaranteed to be satisfied. The configurable products are usually characterized by the following properties [35]:

- each delivered individual product is adapted according to individual requirements
- products are pre-designed in order to accomplish a given range of individual requirements
- each individual product is specified as a combination of pre-designed components or modules
- the products have a pre-designed general structure
- no creative or innovative design is needed as a part of the sales-delivery process
- products are adapted by a routine, systematic product configuration process.

A product configurator is based on configuration software, which is able to map complex configuration rules with or without usage of a geometry kernel to create a CAD model in modular design. A product configurator implements formalized product logic, which contains all “If-Then” configuration rules and constraints. The usability of configurators has been improved also through the use of a compilation step in which the full space of valid configurations can be constructed. Product configuration tasks are shown in Fig. 14.7. The customer inputs his detailed requirements controlled by the user interface. The configuration problem consists in finding a feasible product that satisfies both the customer requirements and feasibility constraints. A configurable product is defined by a set of attributes or components whose possible values belong to a finite set of values and a set of constraints on these attributes, which specify compatible combinations of the values. A product, which meets the customer’s requirements in the best way, is then selected. After validity check and cost analysis, the bill of material (BOM), CAD models, and finally, the bid are generated.

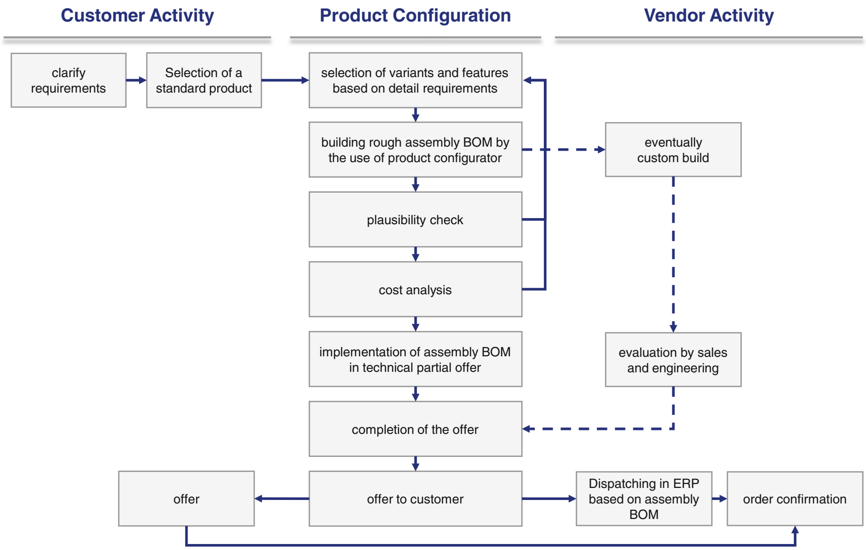

Customer Activity
Product Configuration
Vendor Activity
clarify requirements
Selection of a standard product
selection of variants and features based on detail requirements
building rough assembly BOM by the use of product configurator
eventually custom build
plausibility check
cost analysis
implementation of assembly BOM in technical partial offer
evaluation by sales and engineering
completion of the offer
offer
offer to customer
Dispatching in ERP based on assembly BOM
order confirmation

Fig. 14.7 Product configuration system

Today available commercial configurators have been designed as either enterprise resource planning (ERP)-centric or CAD-centric, depending on the need for a final CAD design activity (e.g. mounting specific adjustments) after the configuration is closed. By force of circumstance, as its function affects multiple core areas of an enterprise, a product configurator has to be integrated deeply with the involved IT systems such as ERP, product lifecycle management (PLM), CAX technologies. However the complexity associated with managing and synchronizing configuration master data across different applications such as ERP, PLM and CAX is an important barrier to the deployment of integrated product configuration. One study gives a comprehensive overview of the product configurators that are available on the market [36]. This plethora of commercial solutions indicates the maturity of this approach to support modular design as well as its deep market penetration.

14.4.3 Agent-Based Approach

Design for configurations is not only a structural problem but also a collaborative design problem [37]. Product configuration must explicitly consider different actors and their perspectives influencing the design of configurable products simultaneously. Uncertainty is also an integral part of configurable product modeling. Indeed, configurable product modeling must be able to deal with various unstable and imprecise requirements coming from customers, on the one hand, and some forms of uncertainty such as imprecision, randomness, fuzziness, ambiguity, and incompleteness, on the other [38]. Imprecision is caused by non precise or exact nature of design information. Randomness is referred to the lack of predictability or irrelevant or meaningless data considered as noise. Fuzziness is caused by incapacity to define the semantic of a variable rigorously, or the definition could be meaningless. Ambiguity is caused by indefiniteness in several interpretation of one word in the same language. Incompleteness is caused by the lack of information. Design process is, thus, the source of the uncertainty.

The *agent paradigm* can be applied to handle complex *uncertain problems* where global knowledge is inherently distributed and shared by a number of agents, with the aim to achieve a consensual solution in a collaborative way. Agents are autonomous and distributed entities capable of executing tasks either by themselves or by collaborating with other agents. An agent is a computer entity, located in an environment that it can perceive, in which it can act, possibly composed of other agents with which it can interact in an independent way. Fuzzy agents are proposed to solve distributed fuzzy problems [39] as well as to model the processing of the fuzziness of information, fuzziness of knowledge, and fuzziness of interactions in collaborative and distributed design for configurations [37, 40]. Structural problems of configuration are also formalized with the help of configuration grammars [41] and implemented in a grammar-based multi-agent platform [42]. An agent-based system called fuzzy agents for product integrated configuration (FAPIC) is developed for product configuration (Fig. 14.8) [43]. In FAPIC, each requirement,

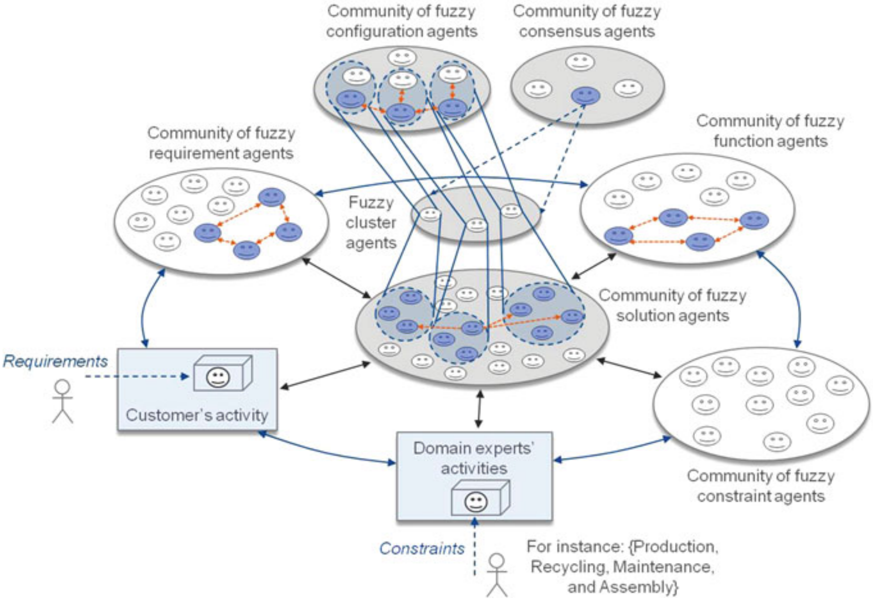

Community of fuzzy configuration agents
Community of fuzzy consensus agents
Community of fuzzy requirement agents
Community of fuzzy function agents
Fuzzy cluster agents
Community of fuzzy solution agents
Requirements
Customer's activity
Domain experts' activities
Community of fuzzy constraint agents
Constraints
For instance: {Production, Recycling, Maintenance, and Assembly}

Fig. 14.8 Agent-based architecture of the FAPIC platform [44]

function, solution and process constraint is a fuzzy agent, with a degree of membership in each community of agents: requirement community, function community, solution community and constraint community.

Cooperative interactions between fuzzy agents are of two types: (1) Intra-communities interactions such as the interactions of fuzzy agents in the intra-community of functions and intra- community of solutions; (2) Inter-communities interaction between fuzzy agents of different communities such as the interactions: (a) between the requirement agents and the function agents, (b) between the function agents and the solution agents, (c) between the process constraints agents and the solution agents, and (d) between the configurations agents and the solution agents. Hence, there are discrete and continuous interactions between the agents of different communities.

In the first phase, FAPIC builds different societies of fuzzy agents, necessary for the configuration of a product. The agent-based system has been built according to five steps: (1) fuzzy agent-based communities building, (2) building interactions between fuzzy requirement agents and fuzzy function agents, (3) building interaction between fuzzy function agents, (4) building interactions between fuzzy function agents and fuzzy solution agents, and (5) building interactions between fuzzy constraint agents and fuzzy solution agents.

In the second phase, the fuzzy set of consensual solution agents emerges. First the *fuzzy set of requirements* for a particular customer is defined. Given the fuzzy set of customer requirements, the fuzzy set of product function agents are computed using the fuzzy relationship between requirement agents and product function agents (*active functions).* As soon as the set of active function agents emerge, solutions agents interact with them to compute the *fuzzy set of solutions*, (*active solutions).* At the same time, constraints of different process views, involved in the configuration task, are defined. After integration of constraints, fuzzy constraint agents interact with active solution agents to converge towards fuzzy sets of solutions satisfying all fuzzy constraints. The fuzzy set of consensual solution agents emerges after the intersection of active fuzzy set of solution agents satisfying customer requirements, fuzzy set of solution agents satisfying all fuzzy constraints.

In the third phase, the optimal configuration emerges from fuzzy consensual solution agents and their affinities. During this phase, the consensual solution agents through their interactions and using their affinities are structured into modules. Maximization of interactions between the consensual solution agents within a module and minimization of interactions of consensual solution agents in-between modules is the objective function to be optimized.

In the fourth, final phase agents seek the consensus. Interactions between fuzzy solution agents and fuzzy configuration agents are built so that the consensus emerges. Thus, consensus agents interact with fuzzy solution agents as well as with the fuzzy configuration agents. They can inform the designer about the different coefficients established to measure the consensus that emerged. Among the advantages provided by discerning the consensus nuclei of configurations by fuzzy agents, is the creation of a common ground for moving toward an acceptable configuration and, thus, it provides assistance to the collaborative and distributed design for configuration. In addition, the fuzzy consensus can promote the capitalization of consensual configurations according to consensual functional requirements and consensual constraints. It permits the expertise of the various actors to be shared.

14.4.4 PDM Approach

In modern PDM systems, the overall structure of a modular product is mapped in a generalized product structure, the so-called “150-percent-bill-of-material” (BOM). Alternative or optional items are initially managed in the database of PDM systems in the same way as all other items, i.e., items as master records with corresponding attributes. Differences to the usual article management arise only in the structuring of the product in the form of bills. Through the use of variants in product structures, PDM systems are able to manage order neutral BOMs with varying and optional positions. This approach is beneficial for product development and less for production and accompanied departments, because there explicit BOMs are needed for each product variant to be produced.

Furthermore, there is a risk that the data management is very complicated, while compromising the performance of the system needs to be tolerated, especially when a large number of product variants needs to be managed. To resolve these conflicts, modern PDM systems are extended by the module “ariant Manager” of which a schematic workflow is depicted in Fig. 14.9. In the base module all master data (parts, structures and processes) are managed. In case of variants explicit ones are derived by the configuration and clone modules. Various reports can be generated by a reporting module that also contains neutral data when needed.

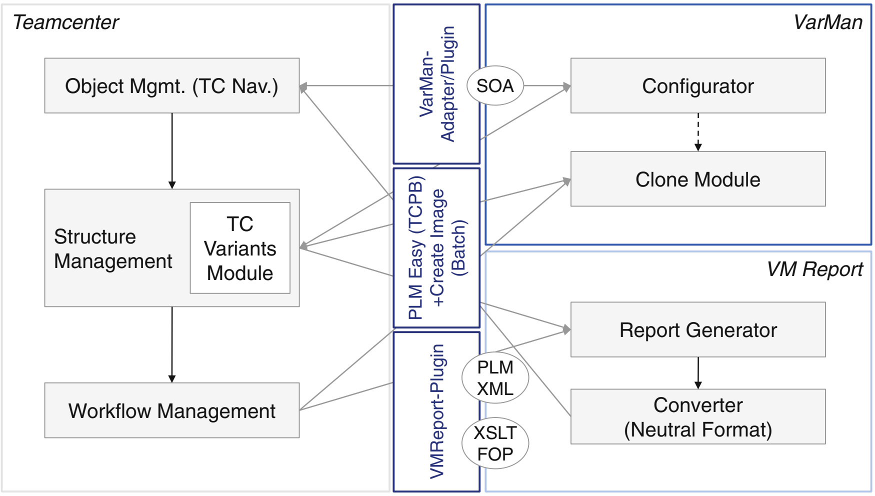


Fig. 14.9 Modular design driven by PDM system

14.5 Use Cases

Several examples of configurable products have been studied in the literature such as: cars, elevators, computer equipment, computer software, telephone switching systems, telecommunication networks, etc. Many companies such as Siemens [44], Mercedes [45], Jaguar Land Rover [46], Volvo Trucks and Fiat [47] have their own history in the development of configuration technologies and tools. This chapter is completed by three use cases, which demonstrate the support of modular design by appropriate configuration tools in the automotive and aerospace industry, and plant manufacturing.

14.5.1 Automotive

Jaguar land rover (JLR) is a premium vehicle manufacturer with two brands: Jaguar produces high performance sports cars and saloons, whilst Land Rover produces class leading 4-wheel drive vehicles.

Caused by JLR's varied ownership heritage two product-definition authoring configurators still exist, both originated in the early to mid-1990s. Both configurators can handle "f-Then"rules and they have a constraints capability. Three cooperating IT applications have been designed to enable the company to operate as an assemble-to-order manufacturer: a product definition application, a bill-of-material system and an order-scheduling system. However, configuration data is increasingly to be distributed across multiple applications throughout the enterprise. In the context of assemble-to-order configuration, a use case has demonstrated a range of configuration concepts to be modeled which are representative of the complexity of JLR's vehicles [46]. Based on these experiences JLR decided to introduce a single-rule authoring and configuration management application, which encompasses product configuration in an integral module. Table 14.2 summarises the configuration concepts and configuration principles.

It was found that neither a PLM, nor an ERP-oriented standard application is able to supply the needed functionality for a lifecycle approach to product configuration. PLM systems are product-centered tools, whereas ERP systems consists of operational business tools. A JLR's specific configuration lifecycle management (CLM) system was proposed to support the complete product lifecycle for a JLR configuration [47]. Table 14.3 shows some aspects of PLM, CLM and ERP and their differences from a product configuration management perspective.

The interfacing between PLM, CLM and ERP is conveniently built on product features and their families. Features are the basic building blocks for defining configurations of vehicles and provide the basis for one common modelling language through which to write rules (see Sect. 10.6.3). The features include customer-facing features used in the ordering of products as well as technical features, which are unimportant to customers but are essential in driving the manufacturing

Table 14.2 Configuration concepts and configuration principles [46]

Configuration concepts tested	Configuration principles
1. Support for effectivity	• Support for effectivity based configuration throughout the product lifecycle.
	• Effectivity is a time point or set of points which define the availability period of an object e.g. a feature, a feature rule, a feature specification, etc., in a product.
2. Support for concurrent build periods	• Differentiation between production builds and prototype builds using effectivity.
	• Releases managed around timing points and change points, with multiple programms of work being managed independently.
3. Support of an integrated modelling environment	• Support for both perpetual and model year based releasing in the product-modelling environment.
4. Support for rule inheritance	• Creation of marketed intent will be defined within the bounds defined by the engineered intent.
5. Support for the authoring cross-carline rules	• Enable modular product modelling, for example, homologation rules and certain architectural technical constraints apply accross multiple car-lines.

Table 14.3 Aspects of PLM, ERP and CLM from configuration management perspective [46]

	PLM	CLM	ERP
Purpose	Supporting product knowledge management.	Supporting configuration lifecycle management.	Supporting operational business requirements.
Approach	Project-based.	Supports both project based and transactional view with one set of configuration master data.	Transaction-based.
Time cycles	Supports time to market.	Supports both time to market and time to customer.	Supports time to customer.
BOM focus	Definition of an engineering BOM.	Generation of solved feature strings.	Execution of a manufacturing BOM.
Solve performance	BOM-explosion workload: 10's per hour.	Performs many types of solves also to support, among other things, BOM explosion at a rate of 10000 s per hour.	BOM-explosionwork-load: 1000 s per hour.
Types of constraints authored	Technical constraints.	Technical and commercial constraints coexist.	Commercial constraints.
Change of product configuration offerings	By model year.	Linking model years and running changes.	Running.
Configuration space	Partially defined configuration space biased towards technical features.	Fully defined configuration space linking technical and commercial features.	Partially defined configuration space biased towards commercial features.

processes. Thus, an important shared resource is a master feature dictionary, which contains the common vocabulary of all allowed features across the involved systems. A master feature dictionary should be managed carefully to avoid duplications of features. The master feature dictionary will evolve over time and some re-modelling of feature families is unavoidable as vehicle technologies develop and otherwise simple features change into combinations of sub-features.

14.5.2 Aerospace

An automated configuration design method and tool for the realistic representation of various aerospace vehicle geometries, using fewer control parameters, build an aircraft geometry step-by-step using typical and common components. It is intended for using them in conceptual and preliminary design phases [48].

This tool, called PCAD, is based on effective use of a super-elliptic formulation with exponential and polynomial distribution functions in an example of configuration design. With this design method, designers can represent the geometry either of airplane, helicopter, fighter or missile more realistically, quickly and efficiently. After that, they can store the geometry data in a PDM system, and manipulate complex shapes with a small number of control parameters. An aircraft geometry model in an associative environment, which is represented more realistically and precisely, is applicable to different stages of the design and development lifecycle, and can be edited like any other interactively created CAD model.

PCAD encompasses several modules: a graphical user interface (GUI), an aircraft "configuration scheme selection" decision-making tool, "surface coordinate points generator" ("control points grid"), customized CAD software adapted to the needs of designers and a commercial PDM tool and is fully embedded in CATIA/ Enovia environment by using CATIA V5 CAA interfaces (Fig. 14.10).

14.5.3 Plant Design with Product Configurator and KBE

In plant design, machines with more than 10.000 parts are applied which are documented in 3D-CAD and PDM systems. They are customized by the following criteria: market and customer requirements, technical producibility, own business aims and the general possibility to create modules. Thereby, both arbitrary complexity and the reduction of product offering have to be avoided. The right product configuration is generated by a Web based product configurator. Additionally, a convenient product presentation for given configuration is chosen by using KBE.

Even for factory planning with more than 500 machines in one production hall and internet applications, complex models which show every detail cannot be used for performance reasons. Furthermore, no company wants to share its know-how with its competitors through the internet discovering fully detailed CAD models.

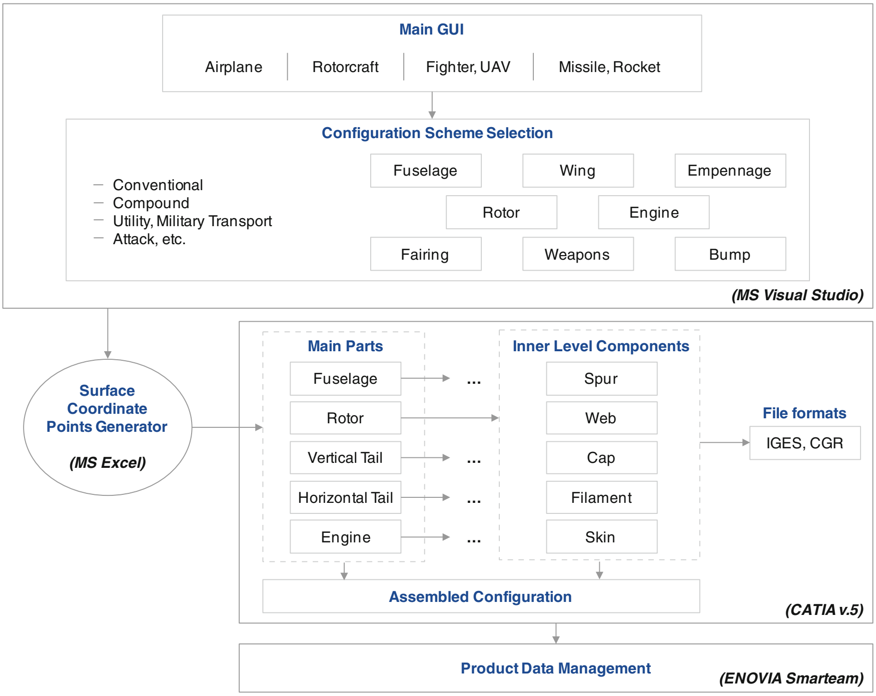


Fig. 14.10 Aircraft configuration design tool [48]

The key is the separation of complex and simplified CAD models in two different data sets, which, though, are managed by the same status information. The simplified model can be generated in different characteristic features (simple, mid, complex). As an example, the sales staff discusses the design of the machine hall with the client and configures the design in 3D on site. In doing so, the characteristics of the individual machines are written down in the CAD system and the simplified models are used. A prime scale drawing can be printed locally and an offer is generated directly. Moreover, it is possible to add a rendered 3D picture to the offer.

In the example of Fig. 14.11 the machine designed by KBE and CAD is able to adapt every of the 50 million possible combinations in the CAD model. According to the client´s choice, the desired variant is adjusted with the product configurator. This variant is checked for doublets at any level of the structure and checked automatically in the PDM system. The parts can be produced directly in the connected sheet bend machine. The variant selection can be effected by an internet solution with the simplified model and, hence, can be directly passed to order management. The processing time for one job is reduced from days to hours.

Configurator can communicate bidirectionally with other sections and internal systems (CRM, ERP, PDM) by detached status information from the CAD system.

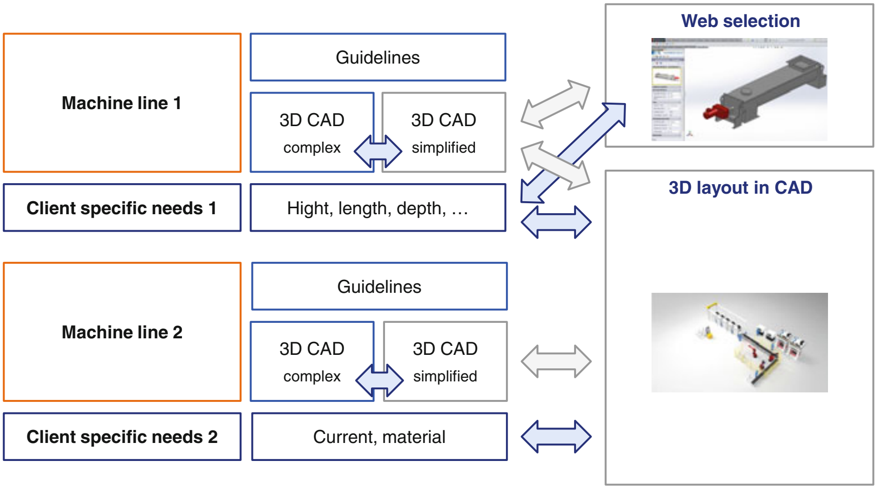


Fig. 14.11 Modular design with product configurator and KBE (Reproduced with kind permission from Lino GmbH © 2014 [49])

This solution allows building up bottom up relationship knowledge and setting up assembly plans by ERP object lists. Similar concepts, which combine product configuration with KBE are used for the design of automotive components [50].

14.6 Further Development

Design for product variety, design for product configuration, and design for mass customization are considered to be highly collaborative and distributed processes. *Therefore, from a holistic point of view, there is still much to be desired in order to achieve system-wide solutions for these design processes and platform-based product development, which can consider collaboration and distribution, intensive interaction between distributed actors, heterogeneity, dynamics and evolution of organization, and the uncertainty.*

The design of a modular product is considered to resolve a system-based interdependency problem. Traditionally, this issue has been seen as system architect's task. Architects design a functional and physical architecture of a system and their greatest concerns are still with the systems' connections and interfaces. The development of modular designs often requires a redesign of the components themselves resulting in new components. Consequently, an architect should assess the achievable technical performance of systems based on their underlying modular or integral architecture. *Modular design should be the result of a coherent and rationale design process, where the options, modular or integral, are early explored in response to technical constraints and the set of requirements. Finding* *the relationship between sparseness, modularity, technical constraints and the set of requirements, could allow such assessment early in design process.* A task in modularity assessment is also the issue of increasing the effectiveness of modularity. *Finding the relationship between the level modularity and the effectiveness of modularity is an open-ended issue.*

Product configuration and modularity are inherently related to product architecture. As the product architecture is considered to be the governing force in lifecycle design, the issue of product architecture lacks theoretical foundation. The design of product architecture has been considered rather more as a know-how issue of architects than a scientific-engineering issue. *In what ways a product architecture, accounting nowadays only for the functional and physical aspects of a modular product, integrate all other lifecycle characteristics is an important issue.*

Actually, the lifecycle of a module is confined to predefined scenarios that depend on its interfaces and its connections. A product with increased adaptability and suitability requires more efforts of design and manufacturing due to increased variety and complexity. *How to design intelligent modules is an important issue related to the design of intelligent products.*

The use of open architecture in modular design is a solution to allow the adoption of new technology. The use of existing modules as well as the use of independently developed modules to design new modular systems, while respecting the integrity of these modules, have to do with the suitability for integration of modules. The adaptability and suitability of modules for integration in a wide range of possibly larger systems is an important issue of the design and development of intelligent systems. *The concept of an intelligent product should maximize the design space of architects and system designers.*

The change management of requirements, functions, solutions and process constraints is another question in modular design. The development of intelligent modular products is strongly related to the development of intelligent models and intelligent tools. Thus, development of intelligent multidisciplinary collaborative and distributed platforms can better handle the modularity and variant management problem. *The multi-agent paradigm has the potential to respond to this challenge and to pave the way for the introduction of innovative technologies in a dynamic environment characterized by important changes and evolution.*

Development of intelligent models and intelligent tools on the one hand and the development of intelligent modular products, on the other, which can communicate and cooperate between them, need holistic and concurrent engineering approaches. *These approaches can offer the possibility of the design of self-sustainable models and self-sustainable products.*

To create long-lived modular systems, the foundations of the system have to reflect the corresponding relevant reality. The design of a modular product should exploit this principle thoroughly. More modularity is better in all lifecycle viewpoints. However, except architects, other actors like development project team members and management in general have often limited access to dependency-based system views. *Transfer and sharing of knowledge, from architect to various actors and vice versa, are essential to be able to support all lifecycle viewpoints in* *system level project coordination. If collaborative design in this context is to be successful, it must be built on a shared rationale of critical design decisions.*

A key motivation of modularity is the specialization in the design and production of modules. The modular product serves much larger user groups over longer periods of time than a single combined product. Thus, the performance of the structure of modular product reflects the performance of actors' coordination in an organization. Should a modular organization in a *dynamic world* reflect the modularity of the product, and, should a modular product reflect the modular organization, are still open questions. *Thus, finding the relationship between the performance of the structure of modular product and the performance of coordination of an organization could allow the early assessment of modular product design.*

14.7 Conclusions and Outlook

Modules are encapsulated groups of similar interconnected physical components that operate on a flow of energy, material, or information and perform a set of functional requirements. Autonomy or independence from external, dependence of internal is an important characteristic of modules. Minimization of interactions with external components and maximization of interactions between components within the module is an important characteristic of modules. The common attributes of modular products are: commonality of modules, combinability of modules, function binding in modules, interface standardization between modules, and higher interaction within module versus lower interaction in-between modules.

Modularity is an important property of product design as a multidisciplinary concept. In the context of concurrent engineering methods, modularity can be defined as the degree to which a product's architecture is composed of modules to respond to a set of requirements, including lifecycle issues and the organization of collaborative and distributed design processes. It includes also the organization of quantities of data, information, and knowledge of these design processes.

Design methods for modular design use principally the decomposition principle. The decomposition of functions into ordered sub-functions such that a group of ordered sub-functions can be modularized, the decomposition of relationship between components such that a cluster of components can define a module, as well as the decomposition of the relationship between the function and components such that a cluster of components corresponds to a cluster of functions, are the most used modular design approaches. Methods such as MFD use these principles from requirement to production. The goal is to facilitate the development of a "robust" production program adaptable to the varying requirements. Modularity is also used as a key measure of design for product variety, design for product configuration and design for mass customization. To optimize modularity the Axiomatic Design, the Design Structure Matrix, and the advanced variant mode and effects analysis (VMEA) can be used.

Various technologies and tools have been developed and used to achieve system-wide solutions for modular designs. The current trend is to use, combine and integrate different technologies such as advanced CAD systems, product configurators, agent based systems and PDM systems. Development of intelligent models and intelligent tools as well as the development of intelligent modular products (i.e. intelligent model-tool-product system), which can communicate and cooperate, demands the design of more intelligent organizations of designs processes for product variety, for product configuration, and for mass customization. Development of intelligent model-tool-product systems needs the development of holistic and concurrent engineering approaches. These approaches can offer the possibility of the design of intelligent self-sustainable models and intelligent self-sustainable products.

References

1. Manovich J (2001) The language of new media. MIT Press, Cambridge
2. Fodor JA (1983) The Modularity of Mind. MIT Press, Cambridge
3. Salvador F (2007) Toward a product system modularity construct: literature review and reconceptualization. IEEE Trans Eng Manag 54(2):219–240
4. Sosa ME, Eppinger SD, Rowles CM (2007) A network approach to define modularity of components in complex products. J Mech Des 129(11):1118–1129
5. Boysen N, Scholl A (2009) A general solution framework for component-commonality problems. Bus Res 2(1):86–106
6. Jiao JR, Simpson TW (2007) Siddique Z (2007) Product family design and platform-based product development: a state-of-the-art review. J Intell Manuf 18:5–29
7. Baldwin CY, Clark KB (2006) Modularity in the design of complex engineering systems. In: Braha D, Minai AA, Bar-Yam Y (eds) Complex engineered systems—science meets technology. Springer, Berlin, pp 175–205
8. Kuntz L, Vera A (2007) Modular organization and hospital performance. Health Serv Manag Res Royal Soc Med Press 20(1):48–58
9. Baldwin CY, Clark KB (1997) Managing in an age of modularity. Harvard Business Review 75(5): 84–93. Harvard Business School Publ. Boston
10. Pahl G, Beitz W, Feldhusen J, Grote KH (2007) Engineering design a systematic approach, 3rd edn. Springer, London
11. Priest JW, Sanchez JM (2001) Product development and design for manufacturing: a collaborative approach to producibility and reliability, 2nd edn. Marcel Dekker, New York
12. Ehrlenspiel K, Kiewert A, Lindemann U (2007) Cost-efficient design. Springer, Berlin
13. Ong SK, Xu QL, Nee AYC (2008) Design reuse in product development modeling, analysis and optimization. World Scientific Publishing, Singapore
14. Gawer A (2009) Platforms, markets and innovation. Edward Elgar Publishing, Cheltenham
15. PillerFT Tseng MM (2010) Handbook of research in mass customization and personalization. World Scientific Publishing, Singapore
16. Fogliatto FS, da Silveira GJC (2011) Mass customization: engineering and managing global operations. Springer, London
17. Parry G, Graves A (2008) Build to order: the road to the 5-day car. Springer, London
18. Hvam L, MortensenNH Riis J (2008) Product customization. Springer, London
19. Hüttenrauch M, Baum T (2008) Effiziente Vielfalt Die dritte Revolution in der Automobilindustrie. Springer, Berlin

20. Junge M (2005) Controlling modularer Produktfamilien in der Automobilindustrie. Deutscher Universitätsverlag, Wiesbaden
21. Wei G, Qin Y (2012) Framework of rapid product cost estimation based on the modular product family. In: Chen R (ed) Proceedings of 2011 international conference in electrics communication and automatic control. Springer, London, pp 9–14
22. Rapp T (2010) Produktstrukturierung: Komplexitätsmanagement durch modulare Produktstrukturen und –plattformen, 2nd edn. Books On Demand, Norderstedt
23. Schuh G, Arnoscht J (2012) Aleksic S (2012) Systematische Gestaltung von Kommunalitäten in Produkten und Prozessen. ZFW, Jahrg 107(5):322–326
24. Daniilidis C, Enßlin V, EbenK, Lindemann U (2011) A classification framework for product modularization methods. ZFW, Proceedings of the 18t^h international conference on engineering design, ICED11
25. Suh NP (2001) Axiomatic design: advances and applications. Oxford University Press, Oxford
26. Holmberg G (2002) A modular approach to the aircraft product development capability. In: 23r^d congress of international council of the aeronautical sciences, 8–13 Sep, 2002, Toronto. http://www.icas.org/icas_archive/icas2002/papers/652.pdf. Accessed 15 July 2013
27. Hirtz J, Stone RB, McAdams DA, Szykman S, Wood KL (2002) A functional basis for engineering design: reconciling and evolving previous efforts. Res Eng Des 13:65–82
28. Stone RB, Wood KL (2000) A functional basis for engineering design: reconciling and evolving previous efforts. Des Stud 21(1):5–31
29. Eppinger SD, Browning TR (2012) Design structure matrix methods and applications. MIT Press, Cambridge
30. DSMweb.org. http://www.dsmweb.org/en/dsm.html. Accessed 15 July 2013
31. Cheng Q, Zhang G, Gu P, Shao X (2012) A product module identification approach based on axiomatic design and design structure matrix. Concurrent Eng 20:185
32. Erixon G (1998) Modular function deployment–a method for product modularisation, PhD thesis, The Royal Institute of Technology, Stockholm
33. Bergman B, de Mare J, Loren S, Svensson T (2009) Robust design methodology for reliability: exploring the effects of variation and uncertainty. Wiley, Chichester
34. Fischer JO (2008) Kostenbewusstes Konstruieren: Praxisbewährte Methoden und Informationssysteme für den Konstruktionsprozess. Springer, Berlin
35. Tiihonen J et al (1996) State of 10 cases in the Finnish industry. In: Tomiyama T, Mäntylä M, Finger S (eds) Knowledge intensive CAD. Chapman & Hall, London, pp 95–114
36. Brinkop A (2013) Marktführer Produktkonfiguration, Version 1.25, 4. Juli 2013. http://brinkop-consulting.com/guide/marktfuehrer.pdf. Accessed 15 Oct 2013
37. Ostrosi E, Fougères A-J, Ferney M (2012) Fuzzy Agents for product configuration in collaborative and distributed design process. Appl Soft Comput 12(8):2091–2105
38. Deciu ER, Ostrosi E, Ferney M, Gheorghe M (2005) Configurable product design using multiple fuzzy models. J Eng Des 16(2–3):209–235
39. Munoz-Hernandez S, Gomez-Perez JM (2005) Solving collaborative fuzzy agents problems with CLP (FD). LNCS 3350:187–202
40. Ostrosi E, Fougères AJ (2011) Optimization of product configuration assisted by fuzzy agents. Int J Interact Des Manuf 5(1):29–44
41. Ostrosi E, Haxhiaj L, Ferney M (2008) Configuration grammars: powerful tools for product modelling in cad systems. In: Curran R et al (eds) Collaborative product and service life cycle management for a sustainable world, Proceedings of the 15t^h ISPE international conference on concurrent engineering (CE 2008). Springer, London, pp 451–459
42. Ostrosi E, Fougères AJ, Ferney M, Klein D (2012) A fuzzy configuration multi-agent approach for product family modelling in conceptual design. J Intell Manuf 23(6):2565–2586
43. Fougères AJ, Ostrosi E (2013) Fuzzy agent-based approach for consensual design synthesis in product configuration. Int Comput Aided Eng 20:259–274
44. Falkner A, Haselböck A (2012) An overview of configurator use at Siemens, 2012 Oxford Configuration Workshop, 12–13 Jan, Oxford University. http://www.cs.ox.ac.uk/OxfordConfiguration/. Accessed 15 Oct 2013

45. Kübler AJ, Zengler C (2010) Model counting in product configuration. Workshop on logics for component configuration (LoCoCo 2010) EPTCS 29, pp 44–53
46. Batchelor J, Andersen HR (2012) Bridging the product configuration gap between PLM and ERP—an automotive case study. In: $19t^{h}$ international product development management conference, Manchester, 17–19 June 2012
47. Helo PT, Xu QL, Kyllonen SJ, Jiao RJ (2010) Integrated vehicle configuration system—connecting the domains of mass customization. Comput Ind 61(1):44–52
48. Azamatov A, Lee JW, Byun YH (2011) Comprehensive aircraft configuration design tool for integrated product and process development. Adv Eng Softw 42(2011):35–49
49. Lino GmbH. http://www.lino.de/3d_layout.html. Accessed 15 Feb 2014
50. Elgh F (2014) Automated engineer-to-order systems A task oriented approach to enable traceability of design rationale. Int J Agile Syst Manag 7(3–4):324–347