palgrave macmillan

Product Information Management for Mass Customization

Connecting Customer, Front-office and Back-office for Fast and Efficient Customization

Cipriano Forza and Fabrizio Salvador

Product Information Management for Mass Customization

This page intentionally left blank

Product Information Management for Mass Customization

Connecting Customer, Front-office and Back-office for Fast and Efficient Customization

Cipriano Forza and Fabrizio Salvador

palgrave
macmillan

First published 2006 by
PALGRAVE MACMILLAN
Houndmills, Basingstoke, Hampshire RG21 6XS and
175 Fifth Avenue, New York, N.Y. 10010
Companies and representatives throughout the world

PALGRAVE MACMILLAN is the global academic imprint of the Palgrave Macmillan division of St. Martin's Press, LLC and of Palgrave Macmillan Ltd. Macmillan® is a registered trademark in the United States, United Kingdom and other countries. Palgrave is a registered trademark in the European Union and other countries.

ISBN 13: 978–0–230–00682–9 hardback
ISBN 10: 0–230–00682–5 hardback

This book is printed on paper suitable for recycling and made from fully managed and sustained forest sources.

A catalogue record for this book is available from the British Library.

Library of Congress Cataloging-in-Publication Data

Forza, Cipriano, 1962–
Product information management for mass customization : connecting customer, front-office, and back-office for fast and efficient customization / by Cipriano Forza and Fabrizio Salvador
p. cm.
Includes bibliographical references and index.
ISBN 0–230–00682–5 (alk. paper)
1. Product management. I. Salvador, Fabrizio, 1971– II. Title.

HF5415.15.F67 2006
658.5′1–dc22 2006043218

10 9 8 7 6 5 4 3 2 1
15 14 13 12 11 10 09 08 07 06

Printed and bound in Great Britain by
Antony Rowe Ltd, Chippenham and Eastbourne

To Lara and Nicoletta,
our loving wives and
patient companions in the
journey of life

This page intentionally left blank

Contents

List of Figures

List of Tables

List of Cases

Preface

The history of product configuration started several years ago, when the Digital Equipment Corporation gathered a team of ‘knowledge engineers’ with an apparently impossible mission: to create a system capable of helping salespeople in the complex definition of microcomputer PDP specifications, and of supporting production technicians in the definition of the components needed to satisfy such specifications. The problem arose from the fact that PDP microcomputers could be built in millions of different variants, taking into account hundreds of technical constraints. Consequently, the Digital Equipment Corporation was forced to invest considerable resources in customizing its products. The experience of this company was a pioneering attempt to gather and formalize the necessary knowledge to customize a product. The contribution of these pioneers, as usual, only produced widespread results a number of years later. Cisco Systems and Dell Computers successfully sold their configurable products thanks to those first ‘knowledge engineers’.

Formalizing the knowledge to customize a product, transferring such knowledge into a product configurator, and realigning the organization to exploit the capability of such product configurator are very complex activities. Unfortunately, no guidelines exist as for the appropriate execution of these activities. On the one hand, those companies which have succeeded in implementing a product configuration system are unwilling to disclose their successful ‘recipes,’ as they try to protect their competitive advantage. On the other hand, most of the texts dealing with this subject do not tender advice to managers, as most texts are typically addressing an academic or overly specialized audience. Last but not least, extant literature on product configuration is dominated by a technocratic view of these technologies, and fails to recognise the organizational concerns associated with the implementation of a product configuration system.

Whilst there are only a few companies that have a clear idea of how to face the problem of product configuration, there are many others in great need of answers, of the know-how needed to tackle the obstacles. Unfortunately, the only easily accessible source of information comes from software houses which sell product configurators. It is evident that this information is focused on commercial purposes and does not follow the systematic and objective criteria which are typical features of scientific works. This volume aims at formalizing the principles according to which a company may offer customized products, following product configuration approaches. In addition, this work introduces a type of language to describe this particular subject, establishing the bases for scientific management research in this topic. Finally, it discusses the mechanisms through which principles are translated into operational performances, encouraging the company to reach the goal of 'efficient customization'.

From a methodological point of view, this work is the synthesis of research that lasted for four years. Case studies and interviews with configuration system programmers, system engineers, managers, executives and consultants allowed a holistic approach to product configuration problems. Research-intervention activities helped us to immediately understand the real problems related to the adoption of this approach. Finally, a study of a vast number of enterprises that offer product customization persuaded us of the extent and relevance of the problems analysed in this volume. We hope the answers this book supplies will be equally appreciated.

The results of this research cannot be attributed solely to the authors: a good number of other people contributed to clarifying the subject. There are too many to mention individually, but we offer our heartfelt thanks to all those who have offered their valuable experience and knowledge.

Financial support for this book was granted by the Spanish Ministry of Science and Education, National R&D Plan, (Ministerio de Ciencia y Educación, Plan Nacional de I+D), Project #SEJ-2004-08176-C02-01.

Cipriano Forza
Fabrizio Salvador

Reader's Guide

There are twelve chapters in this book divided into four parts. Part I introduces the subject of product configuration. In particular, Chapter 1 highlights the field where the concept of product configuration is applicable. Chapter 2 defines the process of product configuration and configurable products. Finally, in Chapter 3 traditional approaches to product configuration are discussed, describing their advantages and disadvantages.

Part II illustrates the structures and functional mechanisms of configuration systems supported by product configurators. Chapter 4 explains how to integrate organization and information system, distinguishing situations characterized by different degrees of automation. Chapters 5 and 6 analyse in detail the logic applied to formalize the knowledge needed to carry out product configuration, from the commercial and technical-production points of view respectively. Chapter 7 offers a general view of the models underlying the automatic generation of codes, costs, prices and graphic outputs for configurable products.

Part III deals with the selection of a product configurator, its implementation and links with other parts of the company's information system. In particular, Chapter 8 analyses how the configurator interfaces with the production planning system and the advanced information system, especially useful in the presence of customized products, such as Product Data Management and Customer Relationship Management. Chapter 9 offers a detailed picture of the various aspects which differentiate product configurators while Chapter 10 presents a structured approach to the implementation of the product configuration system.

Finally, Part IV analyses the impact of configuration systems on the company as a whole. Chapter 11 presents a detailed case study of an enterprise that successfully implemented a configuration system, describing the changes in operational processes and in the related services. Chapter 12 describes how the implementation of a configuration system affects the whole organization, and how 'customizing using configuration' requires an integrated action on the product, the organization and the information systems that support product knowledge management.

How to read this book

This book is addressed to different kinds of readers. According to the type of reader, there are various itineraries to follow while reading this work, that will satisfy specific cognitive needs with the least possible effort.

- University students: Engineering or Business Management students, who are attending a special course on product configuration or on product variety management, can read this book following the order of the table of contents. To those who study Engineering and are attending, for example, a basic course on Management, we suggest reading Chapters 1, 2, 3, 11 and 12 as complements to basic managerial knowledge.
- Directors, managers and consultants: this category of readers includes people with different roles. A company director will find it useful to read Chapters 1, 2, 4, 10, 11 and 12. Managers of different areas may be interested in reading more specialized subjects, such as Chapters 5, 6 and 7. The person in charge of information systems should consider the whole book, with particular emphasis on Chapters 8, 9 and 10. The same criterion is valid for management consultants and system engineers of software houses who implement configuration systems.
- Researchers: this book offers a wide variety of themes, therefore researchers from different fields will find it useful. Obviously, those who carry out research into product configuration can read this work following the table of contents. Those interested in mass customization, product variety management, new product development and business process re-engineering should consider Chapters 1, 2, 3, 11 and 12. Finally, to researchers on knowledge management we suggest Chapters 11 and 12, possibly complemented by Chapters 2, 4, 5 and 6. Chapters 5, 6, 7, 8 and 12 may be of interest to those who study e-commerce.

Part I

Product Configuration

This page intentionally left blank

1

Product Configuration: a New Approach to More Efficient Product Customization

Product configuration is nowadays particularly relevant in relation to product customization, which is a well-established trend. The aim of this chapter is to set the scope of product configuration, illustrating why and when this topic is relevant for a company. In doing so product configuration is linked to the changes in the contemporary competitive arena. More specifically, this chapter:

- illustrates the trends of product proliferation and 'cost-variety' trade-off reduction;
- explains the relationship between product variety and customization;
- relates product configuration with product variety and product customization.

1.1 The trend towards product proliferation

The increase of product variety offered by enterprises is, no doubt, one of the main characteristic trends of modern economic systems. In spite of the homogenization of the individual needs and the liberalization of international trade, related to globalization, this trend is found in several areas. Considering the American market from 1970 to 1998, for example, there was a systematic increase of product variety in different sectors: automobiles, sportswear, electronic devices, food etc. (see Table 1.1).

There was an increase in the variety of products offered by different manufacturing sectors as well as in the products made by each single enterprise. This is true not only for the companies that manufacture consumer goods, but also for those that produce durable goods and commodities, as shown in Figure 1.1.

Table 1.1: Trend in product variety (no. of models) for some products in the USA

Types of product	*1970*	*1998*
Automobile models	140	260
Newspapers	339	790
TV screen (size)	5	15
Movies (at the cinema)	267	458
Breakfast cereals	160	340
Types of milk	4	19
Mouthwash	15	66
Sports shoes	5	285
Brands of mineral water	16	50
Types of tights	5	90

Source: Cox and Alm, 1998.

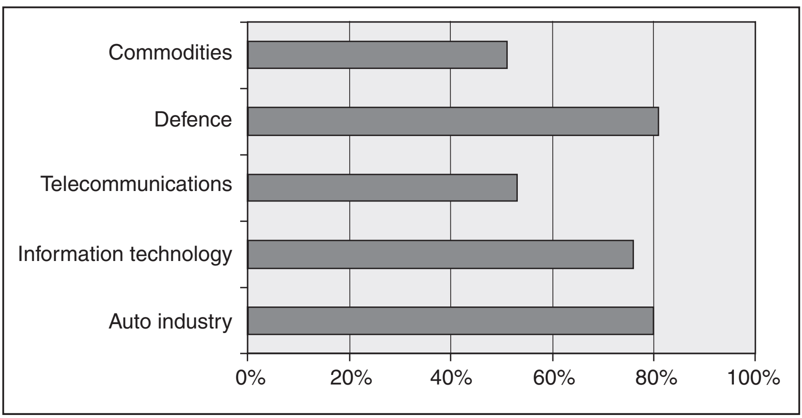


Figure 1.1: Increase in product variety, reported by American enterprises management

Product line extensions are, in fact, an almost compulsory choice for many companies. This decision depends on many factors and on the intensity of their influence. Among the main reasons that influence the development of a wide range of products, four of them are particularly important: (1) market deregulation; (2) product regulation; (3) customer needs and experience; (4) distributors' power and needs. There follows a brief description of the mechanisms by which each one of these reasons helps to determine an increase in product variety.

1.1.1 Market deregulation

The deregulation of many markets and industrial sectors, which in the past were strictly regulated by national laws, is nowadays giving some companies an opportunity to operate at an international level. A standardized, so-called 'global' product would be attractive and advantageous for such companies, above all because they would be able to reduce costs without releasing modified products for different markets. Unfortunately, the presence of environment-specific factors often constrains these companies to tailor their product offerings according to the needs of the new markets they are trying to get into. At the same time, market liberalization not only means a chance for other companies to compete in a given market that was once dominated by a limited number of companies, but it also becomes a mechanism to increase pressure for lower prices, more competitive delivery times, etc. Consequently, companies once protected by strict regulations are now being encouraged to engage in product differentiation in order to escape from the trap of perfect competition. It is useful to remember that market deregulation may affect not only companies operating immediately in the market, but also suppliers to these companies. For example, after the deregulation in the European telecommunication services, telecommunication equipment suppliers had to respond to the demands of greater customization from their customers, i.e. telecommunication service providers. The latter, in fact, forced by greater competitive pressure to cut overhead costs, reduced in-house technical staff. As a result, this decision decreased their ability to adapt equipment prices to the characteristics of their specific telecommunication network, and therefore forced them to transfer these customization activities to telecommunication equipment suppliers.

1.1.2 Product regulation

Cross-national agencies and regulatory bodies that promote standardization, for example the International Standard Organization and the European Commission, etc., are encouraging companies to sell the same product across national markets. Yet, products sold in different countries still have to comply with different regulations and country-specific constraints. To make things worse, we are also witnessing a proliferation of new country-specific regulations (at times regional-specific or city-specific) that mandate compliance with emerging issues according to problems that may vary from area to area, suA(as environmental and safety concerns. For example, natural gas regulators in the city of Vicenza have to comply with the requirements imposed by the local authority that controls gas distribution in the area.

1.1.3 Customer needs and experience

Final customers themselves have been exerting, in recent years, greater demands for more choices over product features. This is due, in part, to the abundance of readily available product information (intensified by the use of the Internet), which has given customers the opportunity to evaluate whether the product meets their specific needs or not. At the same time, being more aware of the multiple available alternatives, customers are less willing to buy a product that does not perfectly satisfy their needs. For example, many customers are now aware that they can ask for refrigerators matching the colour and style of their kitchen, and consequently are less willing to accept a 'standard' white refrigerator.

Moreover, customer experience gained through product use is, itself, a further driver for product customization. In making product replacement decisions, customers can use accumulated knowledge about products, product functions and product usage, to assess the features of different available products in relation to the use they make or are willing to make of the product. For example, when the first-generation microwave ovens were introduced, consumers were primarily concerned with exploring what the product could do. Later on, as experience in usage accumulated, the overall market became more segmented into different customer bases: those who require very basic uses of the oven for warming pre-cooked food or defrosting; those who want a microwave to warm food with a thermal heating function for 'crisp' effect; those who want to combine the function of microwave oven with that of a traditional electric oven for overall food cooking, and so on.

1.1.4 Distributors' power and needs

Finally, distribution channels may represent a further driver that can have an influence on the enormous variety of products requested of different enterprises. The trend, in fact, seems to be towards consolidation of distributors into larger companies capable of reaching a broad base of final consumers. This increase offers distributors stronger bargaining power and the opportunity to demand more advantageous price conditions and to force suppliers to provide differentiated products, for which they have exclusive distribution rights. For example, many mobile phone service providers (e.g. Ornately or Tim) are able to ask cell phone providers, including Nonie and Motorola, to supply variants of the same basic phones that are differentiated in some of their features, such as packaging, software, decals, colours, etc.

But, even in the case of fragmented distributors and retailers, characterized by low bargaining power, they may be motivated to seek differentiation in the products they sell from those offered by competing distributors and retailers. By offering a product that is unique in some features, relative to what is available at the competitors' stores, these distributors can hinder the customer's ability to engage in direct comparison among competing products. For example, in the fragmented Italian retail home appliances business, the availability of product variants that allow the store to offer products slightly different from those offered by nearby competitors, is a key factor in a retailer's decision concerning the assortment of brands he intends to offer.

1.2 Search for an 'efficient' variety

The availability of many types of products, each one in numerous variants, is not a new phenomenon. Handicraft stores have manufactured their products, within certain limits, according to their clients' specifications: lute-makers, tailors, carpenters and so on traditionally manufactured their products incorporating customer specifications. Other economic activities have also traditionally manufactured complex and differentiated tailored products, such as shipyards, construction companies and large mechanical workshops. However, the cost of such flexibility often implied low productivity and long delivery times. In the case of customized production (at industrial or handicraft level), design, supply, manufacturing and delivery must in some way or other satisfy the customer's specific needs. This variability involves an enormous workload in the design activities of the product variants and, at the same time, reduces repetitive operations in the value chain, diminishes productivity and extends the order fulfilment process.

Historically, the answer to operational difficulties associated with tailored products is known as the 'standardization movement'. This movement started in the US at the beginning of the twentieth century and set up the theoretical bases for mass production. It established that, in order to increase efficient design, production and distribution, it is necessary to reduce the impact of customer variability on internal operations, identifying general product and customer categories and simplifying the process of interaction with the customer. These principles allowed a favourable development of economies of scale with successful results, for example, in the automobile industry. Mass production of Ford Model T permitted the company to sell an automobile at $360 (1916) when the average price was over $2000. Many companies, consequently, adopted similar principles, seeking the standardization not only of their products, but also of their design, manufacturing and sales. This trend rapidly spread throughout the industrialized countries.

The reversal of this trend came around the middle of the 1950s, when the concept of segmented markets was formulated. After some years of continuous economic growth, many mass markets were near saturation. Companies were forced to create products that satisfied specific customer needs as well as the 'average' needs of some custumer segments. The transition from segmentation to micro-segmentation, through the identification of smaller and more homogeneous categories, involved many types of products over the following decades. Finally, in the 1990s, the concept of 'mass customization' came about. This concept is based on the idea of satisfying large markets by combining the typical efficiency of 'mass production' with the ability to offer customized products that respond to customer specifications – the typical feature of 'tailored production': in other words, to reach an 'efficient variety' in the presence of large volumes of production. For mass producers, *to seek 'efficient variety' means increasing product variety while maintaining efficiency.*

Mass production was not extended to all originally tailor-produced goods. Many companies that produce tailored durable goods (presses, cranes, air conditioning systems, telecommunication exchanges, etc.) seldom reach large volumes in their production. Therefore, they have to adapt their products to specific customer needs, with a consequent proliferation of product variety. These companies with low production and high customization historically remained out of the mass production process. However, the pressure on prices and delivery times, exerted by international competition, forced these companies to improve their operations without reducing their ability to offer differentiated and often customized products. For these enterprises, *to seek 'efficient variety' means increasing efficiency while maintaining product variety*. The different paths from tailored production to 'efficient variety' are shown in Figure 1.2.

The problem of efficient variety, then, namely how to be competitive while maintaining low production costs, quick delivery times and a vast product variety, is a concern for both mass production and for tailor-production companies.

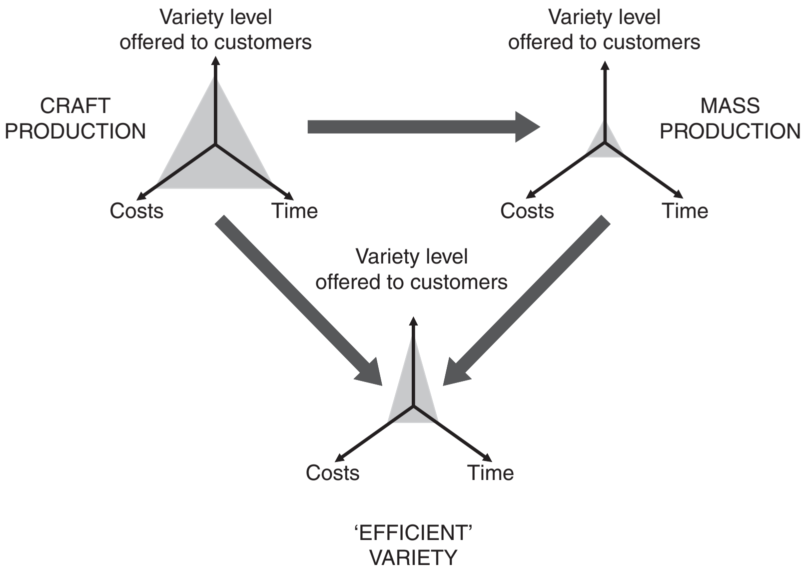


Figure 1.2: Different paths towards overcoming the trade-off between operational performance and product variety

1.3 Variety or customization?

The decision to offer customers a great number of variants of a product is based on the assumption that under these circumstances they are able to appreciate the product utility and then decide to buy it. But the strategies of product proliferation do not always produce the same effect. On the contrary, sometimes they have a negative impact on sales. This negative phenomenon is somehow related to the fact that the potential customer has to weigh up the various features of different variants: he must understand what the company is offering, which features differentiate one variant from the others and has to evaluate how these differences influence the use he is willing to make of the product. Facing these difficulties the customer may postpone the purchase or reduce the costs of selection and buy the product in a shop that offers a more limited assortment. Let us consider, for example, the unpleasant sensation we sometimes feel in a large clothes department store when faced with thousands of clothes on display and where we might have to spend hours looking for a garment that may not be there at all.

A customer buys a product, not only if he can choose among many variants, but also and especially if he has the chance to express his

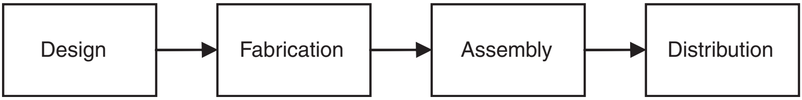


Figure 1.3: Generic sequence of operational activities for a manufacturing company

needs and is then offered a product that satisfies them. In other words, a customized product, instead of a wide range of choices, can be the winning key for order acquisition.

But, what does 'offering a customized product' mean? What fundamental alternatives are available to customize a product? The concept of 'customization' can be explained by considering the four operational activities of a generic manufacturing enterprise: design, fabrication, assembly and distribution (see Figure 1.3).

A product can be defined as 'customized' when one or more activities are carried out according to the customer's specific needs (design

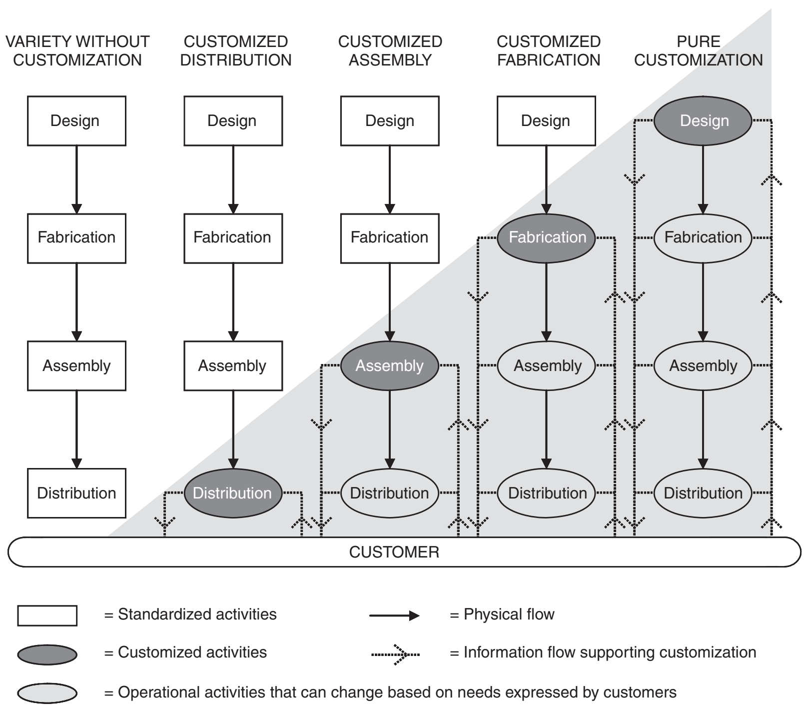


Figure 1.4: Scope of product configuration

and/or fabrication and/or assembly and/or distribution). In other words, we can speak of customization when any of the operational activities responds to a specific feature requested by a certain customer. For example, the nature of customized activities changes according to the customer's opportunities to express his or her preferences about product design rather than about assembly or any other aspect. Therefore, it is possible to distinguish different types of customization, according to which activity, among those illustrated in Figure 1.4, is influenced by customer specifications. There are four different types of customization: (1) pure customization; (2) customized fabrication; (3) customized assembly; and (4) customized distribution. The lowest level is that of a variety without customization.

Pure customization The customer's specific needs are considered from the design process onwards; in this way the product is completely 'made to order'. Producers and customers collaborate to obtain the best solution for both of them. With this type of strategy, all the steps in the production process – design, fabrication, assembly and distribution – are highly customized. Examples of pure customization are offered by enterprises that design and manufacture industrial machinery, building companies, clothing factories and many handicraft shops.

Customized fabrication In this case, customer requirements directly influence the manufacturing activities, not the design process. The company usually offers the potential customers a base product that is later modified according to his preferences. Yet, the degrees of flexibility and the modifications that may change the base product are defined 'a priori'; in this way customer needs are satisfied (within certain fixed variants) without modifying the basic design. This is the case with 'made to measure' kitchens, since the base product fits the size of the room where it will be installed, by simply adapting the wooden board lengths. Similar solutions are adopted for bicycles (e.g. Cannondale) and for industrial tailor-made clothes (e.g. Zegna)

Customized assembly In this case, customer requirements directly influence the assembly activities, not the design and manufacturing process. Products are made with a set of standard components, but the assembly of this set of components is customized to satisfy specific customer needs. This is the case, for example, with customized personal computers (Dell, CHl, etc.) since combining different standard variants of hardware can satisfy the most varied requirements. Another example of customized assembly is the modular bookcase, nowadays offered by many furniture manufacturers.

Customized distribution In this case, the only operational activity influenced by the specific customer is product distribution. The design, fabrication and assembly of the product are not customized. Traditionally, the company manufactures a range of products to satisfy specific market segments. An example of customized distribution could be, among other possibilities, offering the customer a customized list of products including only the varieties that interest him, using the same codes of his information system, including a customized price list and different options for shipping and delivery. Bookshops on-line, such as Amazon.com, offer a typical example. The product (a book) is, undoubtedly, standardized, since it is neither designed, nor manufactured, nor bound according to customer requirements. Yet the customer may specify a series of requirements related to distribution: air, express or ordinary mail, all the books sent in a single package or one by one, invoicing at an address different from that where the books are delivered, and so on.

Variety without customization In this case the customer does not influence any of the activities – design, fabrication, assembly or distribution. This does not mean that the company offers a single standardized product (such as salt or sugar). The enterprise may offer a vast assortment of products and the customer can choose a variant. Many sectors offer 'variety without customization': for example, some furniture manufacturers present a number of solutions, but the customer cannot order a product different from those offered by the company. This is also the case with large supermarkets, such as Auchan and Metro, that offer thousands of products but the choice is limited to what is displayed on the shelves.

The classification of the different types of customization shown in Figure 1.4 has a great advantage since it highlights the difference between *product variety* and *product customization*. The concept of customization is based on the *direct influence customer requirements have* on one or more operational activities (design, fabrication, assembly and distribution).

1.4 Towards efficient customization: product configuration

The distinction introduced in the previous paragraph, between variety and customization, helps us to understand how the company offering

	VARIETY WITHOUT CUSTOMIZATION	CUSTOMIZED DISTRIBUTION	CUSTOMIZED ASSEMBLY	CUSTOMIZED FABRICATION	PURE CUSTOMIZATION
ORDER ACQUISITION	No information related to product customization to be collected	Options and parameters describing product/service attributes have to be collected			New options and/ or parameters describing product/service have to be defined
ORDER FULFILMENT	No customization-related activities	Need for identifying delivery, transportation, etc. Specifications related to each customer order	Need for identifying assembly cycles, part lists, etc. related to each customer order	Need for identifying fabrication specs, set-up, etc. related to each customer order	Ad-hoc design, fabrication, assembly, etc. activities

SCOPE OF PRODUCT CONFIGURATION

Figure 1.5: Scope of product configuration

customized products faces the problem of *linking customer specifications with operational activities* (see Figure 1.5). In order to solve this problem, it is necessary, on the one hand, to *coordinate various departments* so as to verify whether customer needs can be fully satisfied. For example, the engineering department has to check for technical feasibility, the production department has to control manufacturability, quality assurance has to assure compliance to quality standards, etc. On the other hand, linking customer specifications with operational activities requires *'translation' of customer specifications* into all the product information needed by the company to manufacture and deliver the product. For example, customer specifications may have to be translated into design documentation, production cycles, assembly cycles, etc. In other words, product customization generates two problems: how to *coordinate operational activities* and how to *process product information*. The degree of difficulty of the problems is directly associated with the type of customization the company intends to offer. Therefore the number of agents and activities to be coordinated and the volume of information to be processed will increase step by step, if we move from customized distribution towards customized design.

In order to better understand the problems related to different types of customization, it is useful to divide the operational activities into *order acquisition process* (from customer contact to order entry) and *order fulfilment process* (from order entry to delivery).

Order acquisition process In the order acquisition process the company interacts with the customer to communicate what the enterprise offers or is able to offer, how these offerings respond to specific customer needs and under which price conditions, delivery times etc. the transaction is possible. In the case of the *variety without customization* strategy, such interaction is minimized, as it is limited to the quantity of products ordered and to the usual commercial documentation about prices, delivery, etc. In the case of *pure customization*, in contrast, the customer is free to express his needs about everything the company is ready to produce. There is nothing to be chosen because everything must be defined and every imaginable alternative is possible. The customer does not choose a product among all the variants offered, but it is necessary to design a new product different from the 'a priori' possibilities designed by the company. This is generally a time- and resource-consuming process for the technical department. The cases of *customized distribution*, *customized assembly* and *customized fabrication*, represent an intermediate situation. In all these cases, there are a number of options from which the customer can choose, expressing his specific preferences (as far as distribution or technical characteristics of the product). Yet the customer does not affect the product design process as the company has already defined 'a priori' its product families. In other words, there is a problem of product *selection*.

Order fulfilment process The order fulfilment process generates the problem of transferring information regarding the customer specifications to the different inter-company sectors, translating them into appropriate product documentation (lists of components for the assembly process, production cycles, technical features for production, etc.). In the case of *variety without customization*, the problem about how to translate customer specifications does not exist since customer choices do not influence the operational performances. In the case of *pure customization*, the translation of customer specifications affects all the activities and requires more time and resources than in the case of variety without customization. The cases of *customized fabrication*, *customized assembly* and *customized distribution* generate problems related to the information process, which are placed in an intermediate position, between variety without customization and pure customization. These three approaches to customization share a common characteristic: they *do not require* a customized product design. In other words, the product may vary but it is not necessary to engage a designer to set the design parameters that satisfy the specific customer needs.

Analysing the influences that different types of customization have on the order acquisition and fulfilment process, one can find relevant similarities in customized distribution, customized assembly and customized fabrication (Figure 1.5)

- in the order acquisition process: these approaches bring about the issue of selecting among a number of predetermined alternatives, or modifying the parameters that describe the product, within a set of predetermined varieties;
- in the order fulfilment process: the problem is the definition of fabrication and/or assembly and/or distribution activities, associated with a specific customer's order, without involving the product design activities.

The features that distinguish customized distribution, customized assembly and customized fabrication from variety without customization and pure customization, make it possible to gather customization problems associated with these three approaches under the common concept of *product configuration*. To offer product variety through product configuration means giving the customer the opportunity to customize the product while the company is not forced to modify the product design process to satisfy customer requirements.

The product configuration process is a winning element of 'efficient customization' since companies are able to fulfil the highly heterogeneous demands of their customers, offering quick delivery times and competitive prices.

2
Configuration Process and Configurable Products

As product variety increases, product configuration activities grow in complexity, with negative implications on the order acquisition and fulfilment process. In order to avoid this problem, it is necessary to thoroughly analyse the configuration process and the interdependence between configuration process and the characteristics of the product to configure. The aims of this chapter are:

- to define a generic configuration process, identifying the essential logical aspects;
- to explain the different sub-tasks in a generic configuration process;
- to identify the characteristics of configurable products.

2.1 Product configuration process

The information supplied by the customer, regarding the product specifications that better satisfy his needs, is the input for configuration activities. Such information is collected by means of the interaction between the company and the customer. This interaction is fundamentally important for two reasons.

Firstly, the *product offer characteristics defined by the company are communicated*, allowing the customer to explore the variety offered and helping him to find the product variant with the characteristics that best suit his requirements. The more efficient the communication, the greater the opportunities for the customer to decide on his purchase.

Secondly, the description of the product variant resulting from such interaction creates a series of *expectancies* regarding the usage and services of the variant itself. The more accurate the communication of the product characteristics, the higher the customer's satisfaction is as far as product usage is concerned.

Efficiently and correctly communicating the characteristics of a product family chosen by the customer is not an easy task, due to the fact that, quite often, the *language* used by the customer to describe the product and its functions differs, according to the customer, from the language the company uses to describe the same product. It is necessary then, to 'translate' customer needs into product commercial characteristics codified by the company. For example, if two potential buyers of a hi-fi system express their preferences for Chopin or Bach, respectively, a specialized vendor would probably suggest a system equipped with a high peak power amplifier to the first customer, to get more adequate fidelity while listening to the piano's high-pitch sounds, whereas he would propose a system with higher continuous power to maximize low-pitch sounds to the second customer.

Obviously, the language used by the customer to describe the product and the language of the salesman, who directly or indirectly interacts with the customer, are not always very far apart. In some situations the customer has considerable knowledge about the characteristics of the product and is able to describe it 'policode the same language of the company'. Let us take the example of automobiles: it is more common to hear a customer, especially if he is young, asking for 'a 1.9 JTD Stilo with lateral airbags' rather than for 'an economical average-size Fiat that is also fast and safe'. However, when the customer possesses a vast technical knowledge, quite often: (1) there are some differences between the way in which he describes certain product characteristics and the way in which these features are described by the supplier; (2) the customer may use technical language but very often communicates in terms of functionality rather than in terms of the commercial characteristics of the product. For example, a company that installs pumping systems may ask a potential supplier for a submergible axial-flow pump, with a certain capacity, discharge head, overall dimension, with a wiring system according to a certain ISO standard, with minimal maintenance requirements, to be installed in a desert zone. The salesman from the company that produces pumps will not be limited to collecting the apparently complete specifications provided by the potential customer, but knows that he will have to suggest the specific version with plastic impellers, that offers a higher abrasion resistance to sand grains than the option with metal impellers.

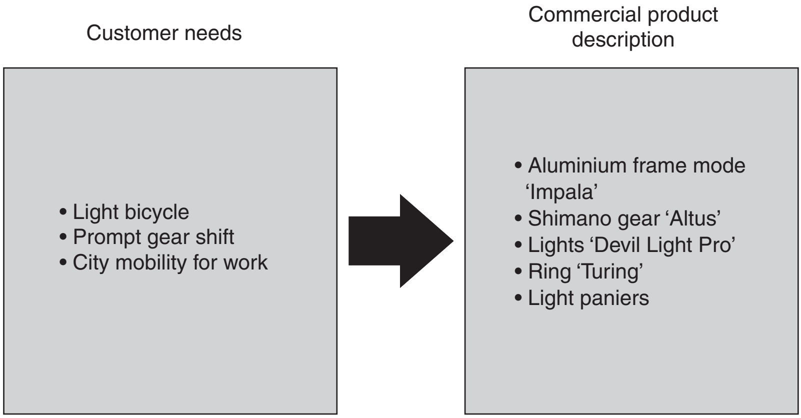


Figure 2.1: From customer needs to a commercial description of the product

When the commercial characteristics to be defined run to dozens or maybe hundreds, as in the case of a loom for textile industry, technical support for commercial configuration activities becomes necessary. The expertise of the technical staff, in fact, is necessary when taking into account the interdependence between all the characteristics to define, or when evaluating the apparently similar alternative product characteristics that better satisfy the specific customer requirements.

In synthesis, the first fundamental step towards product configuration is represented by the 'translation' of the information provided by the customer about the specific product that includes, in different measures: *functionality*, *technical parameters* and even *expectancies*, into a description of all the *commercial characteristics* used by the company to identify a specific product variant. In other words: what is defined as *commercial configuration* (Figure 2.1).

Commercial configuration represents the description of the product the customer is willing to buy and the company agrees to supply. Moreover, in the commercial description of the product some characteristics not strictly associated with the product itself, from the material point of view, can be included, such as price, delivery terms, packaging, etc.

The characteristics that determine a correct commercial configuration are two:

- completeness: all the commercial characteristics of the product are defined. For example, in the case of a voltage transformer with three

secondary turns, maximum powers and currents are specified for the three secondary turns;
- congruence: incompatible characteristics are not defined. For example, in the case of a voltage transformer, the total of the three output voltages from the secondary turns does not exceed the maximum voltage drawn from the primary turn.

Consequently, it is possible to define the *commercial configuration process* as *all the activities carried out to identify the complete and congruent commercial description of the product that best fits customer requirements*.

After obtaining a complete and congruent commercial description on which the customer based his acquisition order, and after completing the order entry, it is necessary to solve the problem of manufacturing the product variant requested by the customer. Generally, the information gathered in the commercial description of the product is not enough to produce the variant. This information reflects what the customer precisely wants, but does not necessarily provide all the indispensable specifications needed for its fabrication. In the case of the voltage transformer, the characteristics of power and current for primary and secondary turns are not relevant to the construction of the electric device: How many copper coils are necessary for each turn? What is the necessary nominal cross-section of the copper wire? The information supplied by the customer does not say anything to the coil-winder operator, nor to the storekeeper who has to withdraw the copper coils from stock, nor to the employee responsible for the supplies, about a possible order of copper wires. In other words, to manufacture the product variant requested it is necessary to have a *technical description of the machine*. When there is great product variety, such description becomes a difficult task, because the parameters that describe the product from the technical point of view depend on parameters that describe the product from the commercial point of view, as in the case of the copper nominal cross-section that is related to the voltage passing through the wire. Obviously, some of these parameters could be independent from commercial choices: for example, the company may have decided to use only one type of steel to manufacture the set of sheets that constitute the nucleus of the transformer, and so the technical characteristic 'type of iron' is indifferent to any preference the customer may express. Finally, in some occasional cases, the commercial characteristics may coincide with the technical ones, thus avoiding the problems mentioned above.

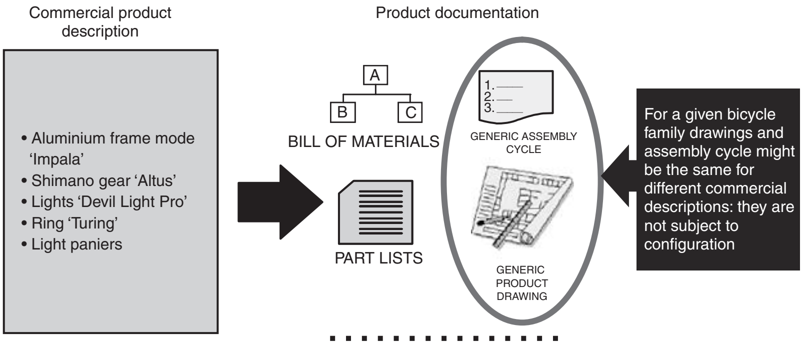


Figure 2.2: From commercial description to product documentation

Once all the technical characteristics of the product have been defined it is necessary to store them and communicate them to the different departments of the company in a practical and understandable format; this information becomes the *product documentation*. The typical product documents are: bills of material, production cycles, lists of acquisition codes, etc. (see Figure 2.2).

Obviously, the nature of these documents, in some measure, depends on the specific type of product and on the company. In the case of a product obtained by the assembly of a great number of components it would be essential to indicate, in the bill of material, how the different parts are assembled to complete the final product, while in the case of products that may require specific manufacturing processes, the production cycle would be fundamental.

It is important to notice that the compilation of technical characteristics in the product documentation has a double aim:

- on the one hand, it is possible to present the 'instructions' in an easy and understandable way for manufacturing the product variant described by the customer;
- on the other hand, it is possible to store the information systematically and in order, i.e. using predetermined logic structures such as: production cycles, bills of material, etc.

The *technical configuration process* can be defined as *all the activities that generate the documentation of the product variant based on the commercial description of such variant*.

Combining the definitions of commercial configuration and technical configuration it is possible to get a general definition of the

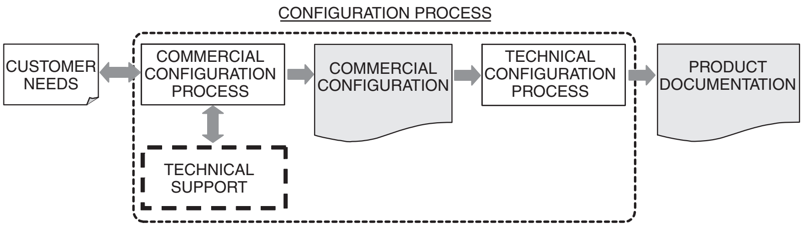


Figure 2.3: Schematic representation of a generic product configuration

configuration process: the *product configuration process* can be defined as *all the activities from the collection of information about customer needs to the release of the product documentation necessary to produce the requested variant* (Figure 2.3).

2.2 Configuration process sub-tasks

As shown in Figure 2.3, the essential logical moments of the configuration process clearly demonstrate the temporal interdependence among them and the nature of the information process. The need to describe a *generic configuration process*, applicable to different contexts and products, imposed the use of a highly simplified model. The specific needs of the company often change and consequently, the sub-tasks necessary to reach the product configuration are likely to change as well, or they may occupy different places within the configuration process. It becomes necessary to complete the configuration process model with the definitions of the different related sub-tasks and their aims.

Communicate possible options to the customer Contact with the customer is the first step of the configuration process. A correct communication of the company's product offer is fundamental to transform such an offer into an order and finally into sales. Effectively communicating the possible options to the customer means giving him the chance to evaluate how the different options may satisfy his/her requirements, thus reducing the cost of selection.

Collect and store customer needs The information supplied by specific customers regarding their preferences represents a resource for the company. Transforming this resource into an intangible beneficial asset requires systematic collection and storage of such information. Analysing this information, it is possible to detect the 'trends of the market' and understand how customer requirements evolve with the passing of time.

Verify completeness and congruence As stated in the previous paragraph, the collection of customer needs must be complete and congruent so as to avoid the necessity of further contacts with the customer to get missing information or to solve inconsistencies found in his choices. To ensure completeness and congruence means defining 'a priori' the product characteristics, pointing out their possible incompatibilities, among which the customer is able to express his preferences.

Supply a description of the product or service (design, chart, texts, etc.) The base of the commercial transaction between the customer and the company is obviously represented by the description of the product or the service to configure. This description includes the *commercial configuration* and could be completed by other types of product descriptions such as: technical designs (see Chapter 13), charts, texts (manuals and electric diagrams for numerical control machines) or even video clips that illustrate the use of the product in typical contexts.

Determine the price The price is obviously one of the key factors that can determine a customer's decision to purchase the product, even if its influence changes according to the context. In the case of a product that can be manufactured in many different variants, it is more difficult to establish the price since each variant requires costs that may vary considerably according to each alternative; consequently, it becomes more difficult to determine the price limit under which it is not profitable to sell a certain product variant.

Determine delivery terms Together with the price, a further component of the feedback supplied to the customer regarding his requirements to conclude the commercial transaction, is represented by the information about delivery conditions. Even in this case, it is evident that if the product presents multiple alternatives, the development of some conditions, for example delivery times, depends on the specific variant chosen by the customer.

Codify the product variant If a company offers a large number of variants it has to tackle the problem of identifying them. In the manufacturing industry this activity is known as product coding. When the object to configure is an intangible product, as in the case of an insurance contract, the activity consists in allocating a record number or a file code.

Select components Once the commercial characteristics of the product have been defined, the necessary product components must be identified. In the case of tangible products, the selection includes the parts and sub-parts that have to be assembled to produce the variant. In the case of an intangible product, such as life insurance, it will be necessary to select all the indispensable paragraphs to include in the contract.

Determine parameters for non-standard components Very often, some of the components that determine the final product configuration are not standard and must be customized, as in the case of the length and height of a window frame.

Determine links between components Once the standard and non-standard components have been determined, it is necessary to establish how to link them. Generally a customer's preferences about different product variants change; consequently the components and the links between them are likely to change as well.

Generate operative instructions to manufacture product variants Once the different components and the links between them have been determined, it is necessary to process this information in a clear and intelligible way for those who actually have to manufacture the product. In the case of tangible products, the typical documents are bills of material, production or assembly cycles, etc., while in the case of intangible products operational procedures are more frequently used.

2.3 'Configurable product' characteristics

From the previous paragraphs it is clear that the configuration process consists in identifying the technical characteristics of the product variant that best satisfy customer needs. The high variability of customer requirements and of technical characteristics makes the sub-tasks of the configuration process uncertain and complex. This complexity becomes time- and resource-consuming, with negative influences on the order acquisition and fulfilment process.

A possible solution could be to establish a priori a clear and certain association among all possible customer needs to be satisfied and the corresponding technical characteristics of the product. In other words, configuration activities may become less complex if any kind of uncertainty is eliminated while 'translating' customer needs into commercial characteristics and then into technical characteristics. To reduce such uncertainty the companies must rationalize their product families in order to offer a 'configurable' product.

But what does 'to rationalize a product family offering a configurable product' mean? The following examples will provide some useful information.

Case 2.1 An example of a configurable product: a moto-variator

In many types of industrial machines it is necessary to modify the rotation cycle of a shaft. Among these machines we find moto-variators, which, among other things, regulate the gear ratio within a certain range of values. Moto-variators may vary from one another in a number of different parameters such as:

- power
- versions (flanged or universal mounting, type of shaft)
- with or without differential (to increase speed)
- type of output configuration (bolt on foot, flanged mounted)
- output shaft diameter
- input configuration
- mounting position, variator position, control position
- type of control, etc.

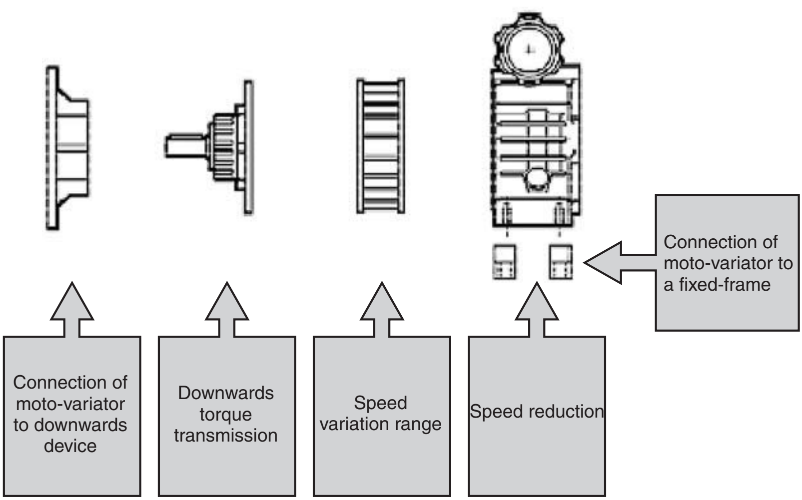


Figure 2.4: Functions of moto-variator components and accessories (adapted from Bonfiglioli.com)

Many of these characteristics are independent from one another, so it is evident that the number of variants may become extremely high. Assuming five options for each characteristic, i.e. five sizes, five versions, etc., we get 6250 variants of moto-variators, without taking into account mounting positions, variator positions, etc. The solution found in the sector is (although there are some important differences among manufacturers) to limit the product family adopting *modular product* architecture. Modular products have been designed in such a way that each of their main components can fulfil one of the product functions and can be freely combined with the other components. In other words, the variator is seen as a system formed by a certain number of components, and each of them can be used in different variants, according to customer needs. For example, if we consider the following functions

- decrease speed
- modify speed
- fix the unit to a frame
- fix the unit to another type of device

it is possible to associate these functions to specific components that are assembled to produce a certain variant of moto-variator with its corresponding accessories.

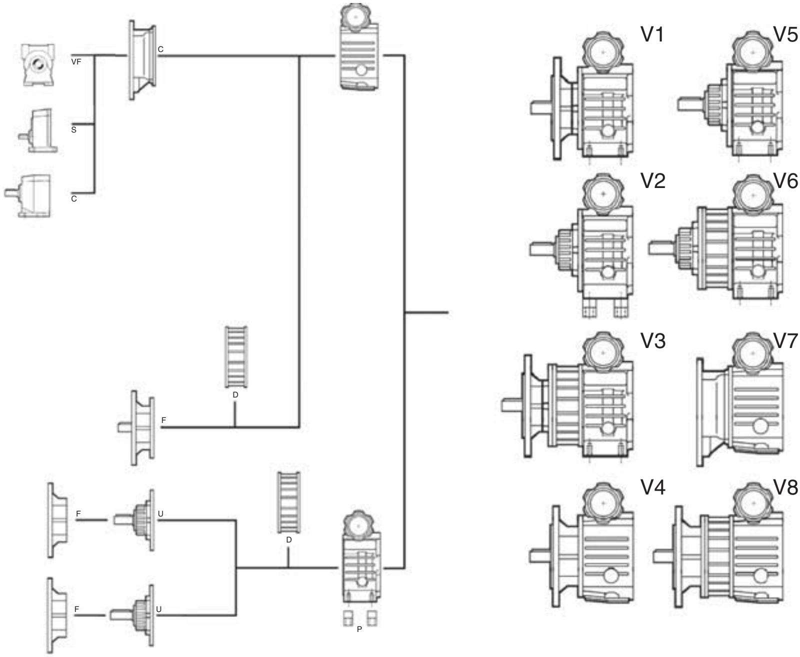


Figure 2.5: Architecture based on the combination of different components (adapted from Bonfiglioli.com)

Once this logic diagram has been elaborated (Figure 2.4), it is possible to make many variants of the moto-variator combining different components. The identification of the necessary components and of the corresponding links becomes a 'mechanical' activity, implicit in the description of the product family. Figure 2.5 is an example of how the different product variants (V1, V8) are obtained combining different types of moto-variator units, flanges and shafts.

The example of the moto-variator suggests that a modular product family is configurable, since it is possible to associate, without uncertainties and without the help of a designer, product functions and customer needs to product components, identifying the way in which these components are combined. By doing so, we can get the necessary 'instructions for building the product' that represent a fundamental output in the configuration process. However, the contrary is not true: a configurable product family is not necessarily modular. We must remember that in the case of a modular product, it is possible to identify a division of the product into components, which correspond one-to-one to the product basic functions. The logic that supports this characteristic of modular products is based on the fact that in order to change a function, it is enough to change only one component, maximizing the similarity of the components among different variants of product. The moto-variator broadly satisfies this requirement. Yet it is not always possible to redesign a product to meet this condition; many products present a complex interdependence between their components and functions and cannot be defined as modular.

Case 2.2 An example of a configurable product: a shirt

Let us consider now a company that manufactures shirts. The shirt represents a typical example of a product that cannot be completely modular. In fact, the differences in measurements among different people require, at least, the definition of different sizes, while differences in taste determine the necessity

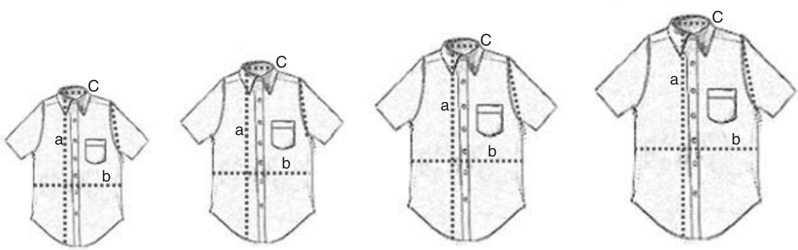


Figure 2.6: Interdependence of technical characteristics of a shirt

of offering different colours, fabrics and finishing (seams, embroideries, etc.). If we consider the question of colours (or fabrics) we immediately see that this single customer's need influences many components that form the shirt: body, sleeves, cuffs, collar. Clearly it is not a modular product. But, worse still, the different builds of the customers require the modification of the sizes, that is to say, a series of parameters that describe all the main components mentioned before. Shirts of different sizes differ, at least in the diameter and length of the body, of the sleeves, collar width and so on. In other words, there is a series of parameters mutually interdependent that vary due to only one necessity, i.e. the shirt should correctly fit the customer (see Figure 2.6).

Although these parameters are interdependent, the relations of interdependence can be codified in rules based on software typically used in the clothes industry, known as grading software. This software automatically elaborates a sort of configuration, based on a model of the piece of clothing designed by the stylist, following some rules that describe how the dimensional parameters of the components change according to the customer's build. Clearly, the nature of such configuration is different from that of the moto-variator. As we have seen:

- all the principal product components change due to the variation of a single customer need;
- the interfaces between components (e.g. collar–body, body–sleeves) change according to variations of customer needs.

The possibility of describing, in a comprehensive way, the interactions between technical parameters (clothes measures, different seams, fabrics, colours, etc.) and customer needs (collar 43, linen, canary yellow) makes the shirt a configurable product. Many companies that produce industrialized shirts 'made to measure' have taken advantage of this characteristic. These companies have increased the control the customer has over the technical parameters, allowing him or her to directly modify a 'standard configuration' of the shirt, by defining sleeve length, waist parameter, shoulder width, etc. (see Figure 2.7).

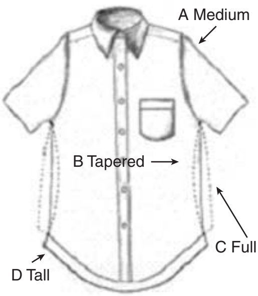


Figure 2.7: Example of technical parameters for a 'made to measure' shirt

These examples clearly show that either for a family of modular products, as in the case of the moto-variator, or for a family of non-modular products, as in the case of the shirt, it is possible to eliminate any kind of uncertainty in the configuration activities, and the product can be defined as *configurable* if:

- all the needs and functions that can be satisfied by the variants of a family product have been comprehensibly defined;
- all the technical parameters associated with such needs and functions have been clearly defined.

The first requirement for the configuration of a product family is a precise definition of what the company is potentially ready to offer and what it is not ready to offer. In other words, the company limits a 'product space' within which the customer will try to satisfy his specific needs. For example, an automobile factory could offer its customers a model with different options, e.g. four different colours for the body, two colours for the inside, three different engines and two gear systems (automatic and manual). This is represented in Figure 2.8, where the polygons drawn on the axes represent the different commercial configurations.

The second requirement is the need to establish a link between the commercial and technical characteristics of the product. The product variants are somehow pre-designed, and their feasibility has been evaluated when defining the family product. In this way it is possible to reduce uncertainties linked to production operations in the configuration process.

Therefore, *configurability* is not a property of the specific product but of the *product family*, defined as a group of products that present similarities in their functions and share the same technology and production process. Using a notion developed in design theory, it is possible to describe a family of configurable products as *a product family whose architecture has been completely defined*. That is to say: a schematic description of the different components that form the product family, of their functions and of the way in which these components interact with one other.

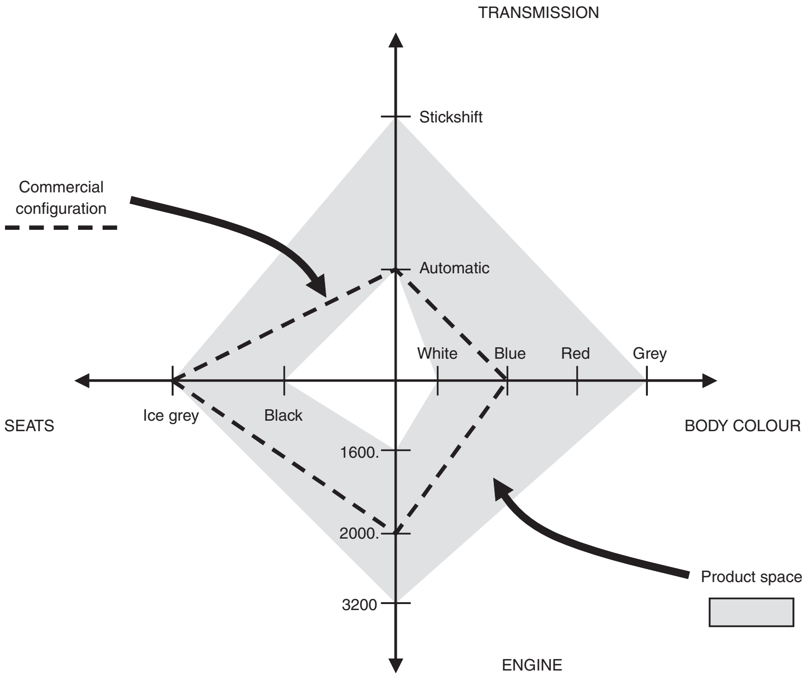


Figure 2.8: Product space and commercial configuration

2.4 Degrees of configuration

Clearly, the complete definition of the architecture of a product family that could be defined as configurable, implies a considerable investment in activities aimed at formalizing the commercial and technical configurations. This investment is not always convenient for the company, especially if a product family presents a degree of configuration below 100 per cent. For example, a company that produces electronic devices could foresee the possibility of the salesman offering input voltage values higher than those used in Western countries, with the aim of expanding the market to developing countries. To cope with this enlargement of the product space, it will be necessary to modify the power adapter. The company will consider this change to be an advantage only if there are concrete opportunities to sell products in the new market.

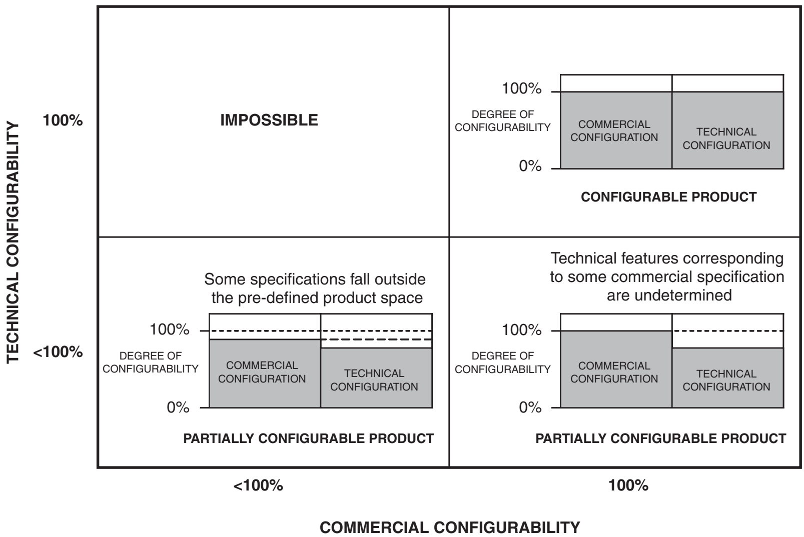


Figure 2.9: Degrees of product configuration

In other words, a possible option is to define a family of products completely configurable from the commercial point of view, but not from the technical point of view. In other situations, a company may be interested in keeping the degree of configuration below 100 per cent, allowing the customer to specify some product characteristics which may be highly unpredictable. For example, many companies that produce components for different types of industrial equipment (gas pressure regulators, pumps, compressors, fans, etc.) offer a range of products structured according to a series of parameters and also leave room to consider all the 'notes' in which the customers ask for special product features, impossible to foresee, such as special paints, tests, identification plates, unusual mountings, etc.

Therefore, we can say that a company does not necessarily have to aim at 'total' configuration. Probably, the company does not have enough volume to get a profitable return on the investment associated to the complete description of the product family architecture, or some of the customer's needs present such a high variability that it becomes impossible and senseless to attempt any kind of anticipation. Figure 2.9 shows a synthesis of the possible situations considering the degree of configuration of a product family.

3
Traditional Approaches to Configuration

In the previous chapter we saw that the order acquisition and fulfilment process often requires an intense information exchange among customer, front-office and back-office. Different approaches were developed in order to simplify this process, spanning totally unstructured approaches, based on loose sheets, to highly structured ones, based on special-purpose configurators.

The present chapter:

- portrays traditional 'paper-based' approaches to product configuration;
- pinpoints typical problems associated with traditional 'paper-based' approaches;
- outlines the evolution of special-purpose configurators, the 'ancestors' of modern product configuration systems.

3.1 Non-structured approaches

The most elementary system used for the collection of customer specifications about a certain variety of the product offered consists in writing down this information on a simple sheet of paper. No doubt, this method does not require any investment to support product configuration. That is why this approach is appealing to some companies, above all to small ones. Moreover, especially in business-to-business companies, salesmen often prefer non-structured approaches, as they appreciate having a free hand in defining and collecting customer requirements. Writing the customer needs on a sheet of paper is an excellent way of demonstrating flexibility and 'pocketing' an order, perhaps starting from one of the models shown in the company catalogue.

Table 3.1: Advantages and disadvantages of non-structured approaches to product configuration

Advantages	*Disadvantages*
✓ No initial investment	✗ Ineffective product assortment communication
✓ High salesman flexibility while bargaining with the customer	✗ Hard to assess validity, completeness and congruence of product specifications
	✗ No support to determine product price

Low investment and great flexibility in business, however, can imply high costs for the company (Table 3.1). This is true, above all, when there is an increase in the number of characteristics offered to the customer, as well as in the number of choices available for each alternative. Under these conditions, it becomes difficult for the sales department employee to communicate the wide range of alternatives offered to the customer by the company. Likewise, it will not always be easy to evaluate accurately whether the customer's specifications are compatible or not. Consequently, there are two possibilities: either to ask the technical office to check if the requested characteristics are feasible, or, in the absence of such control, to run the risk of getting incorrect information from the customer. This situation gets worse if the customer has at his or her disposal a great number of variants to choose from and the salesman possibly forgets one or more specifications: in that case the product configuration may be incomplete. The customer will have to be contacted again in order to gather the missing information, causing misunderstandings and delays. Finally, if the complexity of the product grows, the salesman will find it difficult to determine the product price, being unable to clearly evaluate how customer choices influence the cost and margins.

In spite of what may be thought, many companies – not only small ones – entrust the important duty of collecting customer specifications only to the sales department employee and to a simple sheet of paper, although this difficult task obviously needs other means of support.

Case 3.1 Configuration of an electric transformer using loose sheets

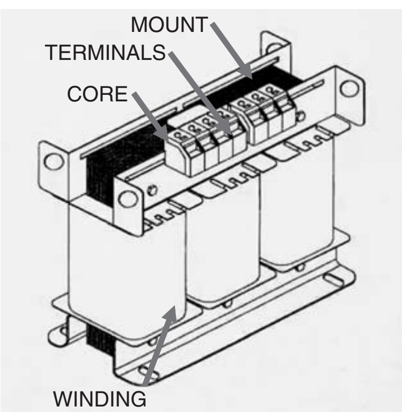
A small company that produces voltage transformers provides a concrete example of the problems associated with the use of non-structured approaches to product configuration. This company (which will be called Blackout Inc.) has progressively abandoned the production of a big series of 'general purpose' transformers to produce customized industrial transformers, to the point of manufacturing more than 70 per cent of the transformers according to customer specifications.

The voltage transformer is a typical example of a configurable product, especially because it has a well-defined architecture: it is made of a set of layered sheets (nucleus) around which copper wires are wound (coil) which are gathered on spools. The electric connection of the machine consists of terminals or other coupling systems, while the mechanical connection with the device or system that houses the machine is formed by sheets or rods integrated with the nucleus (see Figure 3.1). The possibilities for the customer to choose different types of transformers offered by Blackout Ltd. are countless, yet they have to respect the predefined architecture of the transformer: overall power, frequency, winding order, voltage to the primary and secondary turns, current to primary and secondary turns, type of connection (star or triangle), vertical or horizontal assembly, way of connecting to the electric machine, etc. as shown in Figure 3.1.

Transformer customization starts during the order acquisition process, where the properties of the specific product variant ordered by the customer are defined. The sales department, interacting with the customer, has to collect the information necessary to define the attributes of the transformer that can be customized. The sales office staff supply general information on delivery times, cost and product performance. All this information is collected on simple sheets of paper, as indicated in the example shown in Figure 3.2.

The lack of a structured approach when collecting customer specifications leaves the responsibility of this task fully in the hands of the salesman. It is obvious that the sales department must have a certain level of technical knowledge in order to handle the order acquisition process.

The availability of competent technical-commercial personnel, however, is not enough to support the configuration of the electric machine for more complex product families manufactured by the company. In such cases, the



Examples of customizable attributes:

- power
- frequency
- voltage (p/s)
- connection type (p/s)
- mount type
- application
- etc ...

Figure 3.1: Specifications that can be customized in a voltage transformer

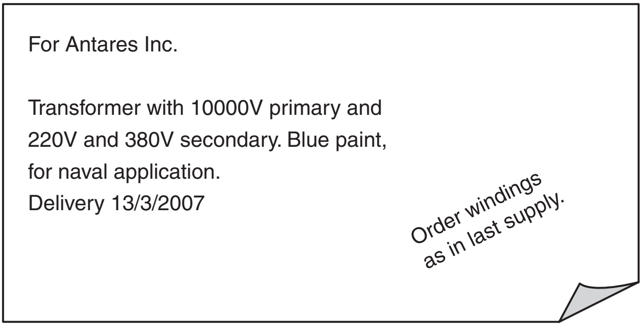


Figure 3.2: Non-structured information of a transformer specification and sales conditions

salesman has to consult the technical department, which advises him about product feasibility and evaluates manufacturing costs associated with the alternatives explored with the customer. The need for the sales department to establish this sort of 'information bridge' between the technical office and the customer causes a considerable delay in the order acquisition times, as well as the risk of misunderstandings on machine feasibility with consequent configuration errors. Finally, these errors can slow down the order fulfilment process, because they have to be corrected by the technical office and because there is a risk of blocking or slowing down production.

The problems associated with the non-structured approach used by the company to customize the electric machine do not finish here. In fact, the salesman faces configuration each time as if dealing with a new variant that has never been produced. Consequently, the sales department inadvertently sends specifications identical to those of machines produced in the past to the technical office. A survey carried out by Blackout Ltd. showed how the company produced around 10 000 product variants in three years. It is clear that things tended to be repeated through the years! The consequences of the difficulty in recognizing a previously produced variant are not irrelevant: the technical office, especially in the case of simple machines, prefers to redesign the transformer, rather than lose time looking for the variant among the thousands of variants filed in the technical archive. This causes a heavy workload for technicians, who can only devote a small part of their time to improving product families or creating new ones.

3.2 Structured approaches to collecting specifications

The easiest way to structure product configuration activities is to define pre-printed or electronic forms that collect customer specifications following a certain format. This is a widespread approach that requires a fairly low initial investment and has some important benefits (see Table 3.2).

Ad hoc forms give the customers a full vision of the properties of the product, from which they can express their own preferences. On the one hand, the customer can rapidly appreciate what the company is offering in terms of customization. On the other hand, showing the customer which choices he can make 'guides' him towards the articulation of specifications that are compatible with what the company can offer and considers advantageous for the customer.

Finally, these specialized forms allow the sales office to check whether the information gathered is complete, and whether all the necessary information for configuration has been specified. Moreover, these forms act as a formal connecting document between the sales department and the technical-production department and help to eliminate ambiguity in the interpretation of the technical specifications that satisfy the customer's needs.

In addition to these advantages there are some important disadvantages that in various ways mirror the former (see Table 3.2). In the first place, even if the specialized forms allow the dimension of the product space to be communicated in an efficient way, they do not give the salesman an 'outline' for progressively guiding the customer in the exploration of this space. Therefore, and especially in the case of complex industrial goods, it is necessary to make use of expert salesmen who know how to lead the customer during the product configuration process.

Secondly, the language used in specialized forms is necessarily close to that used in production, because in general they are considered as connecting documents between the front-office and the back-office. A further difficulty for the salesman is represented by the 'translation' of the product specifications expressed by the customer into the product characteristics from the technical point of view.

Table 3.2: Advantages and disadvantages of specialized forms in product configuration

Advantages	*Disadvantages*
✓ Low initial investment	✗ Hard to assess validity and congruence of product specifications
✓ Completeness of commercial specifications can be checked at a glance	✗ Limited support to determine product price
✓ Fair communication of company's product assortment	

Finally, although the specialized forms describe the product space, they do not help in checking the existence of possible incompatibility in the preferences expressed by the customer. This third disadvantage, in the case of complex products, may become insuperable even for the most expert salesman, limiting the situations in which these instruments could be used successfully.

Case 3.2 Configuring a traditional suit using pre-printed forms

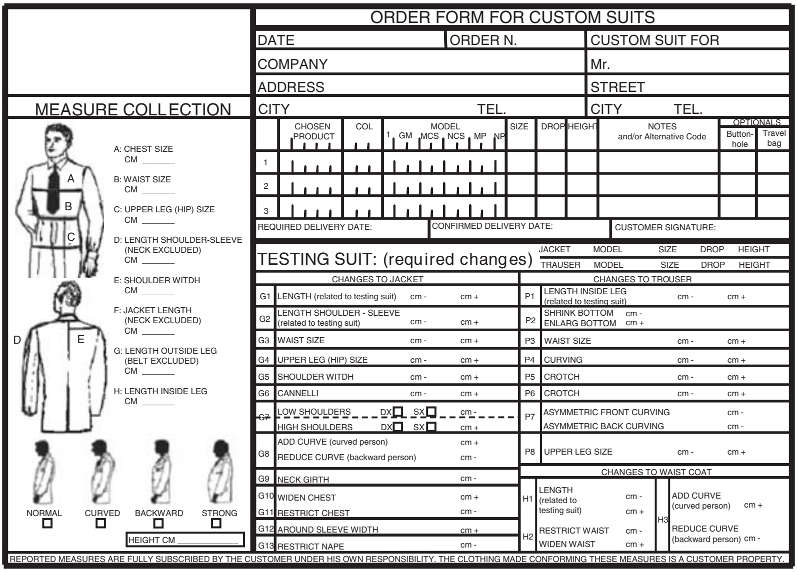
In order to give a concrete example of the problems associated with the use of specialized forms for product configuration, let us consider the case of the configuration of a suit (jacket and trousers)

Mr Ivo Volpon has been working in the fashion sector for more than thirty years and, thanks to his passion for outfit perfection, he has convinced us on several occasions to buy his made-to-measure suits. The problem of tailored garments has always existed: sizes, drop, material and finishing, without mentioning the countless details that distinguish one person from another, make the difference between a personalized garment and a ready-made one. Mr Volpon is convinced that the traditional 'tailor's' approach to the manufacturing of made-to-measure outfits satisfies the absolutely marginal market of those who can and want to spend more than €1000 on a suit. The main method of capturing a 'lower' sector, but one more substantial in terms of volume of the market, is undoubtedly represented by the definition of a series of dimensional and optional parameters regarding material and finishing, that can modify the 'basic' or 'reference' model of the garment.

Mr Volpon says, 'I have to single out a model that can fit the customer's size (sometimes the jacket and the trousers are different sizes) and variant (long or wide) and then personalize the garment on the basis of the morphological characteristics, the posture, the taste of each customer, as well as the occasion when the garment will be worn (work, formal ceremonies, free time, etc.).'

In order to support order acquisition Mr Volpon has developed, in close collaboration with a clothes factory, a pre-printed form – one that is complete and easily understood by the production – that includes all the parameters of the garment that can be customized (see Figure 3.3).

The decision to create this form is due to the fact that the collection of more than forty attributes, on which the customization of a suit is based, cannot be considered an irrelevant task. If a certain parameter is missing, for example, the customer will have to return to the shop, which delays delivery times (usually around two weeks). Alternatively, the production can proceed using a 'default' value of this parameter, running the risk of an imperfect production. In both cases, the risk of disappointing and losing the customer is very high, and no client means no money. It is important to notice that the fact of referring the customization to a precise basic model allows the company that collaborates with Mr Volpon to repeatedly produce made-to-measure suits. They must adjust the cutting operations, according to the specifications directly gathered during the order acquisition. In this way, the price of a made-to-measure suits is just over €500, making the product accessible to a high number of customers.



ORDER FORM FOR CUSTOM SUITS

DATE | ORDER N. | CUSTOM SUIT FOR

COMPANY | Mr.

ADDRESS | STREET

CITY TEL. | CITY TEL.

MEASURE COLLECTION

A: CHEST SIZE CM ______
B: WAIST SIZE CM ______
C: UPPER LEG (HIP) SIZE CM ______
D: LENGTH SHOULDER-SLEEVE (NECK EXCLUDED) CM ______
E: SHOULDER WITDH CM ______
F: JACKET LENGTH (NECK EXCLUDED) CM ______
G: LENGTH OUTSIDE LEG (BELT EXCLUDED) CM ______
H: LENGTH INSIDE LEG CM ______

NORMAL CURVED BACKWARD STRONG

HEIGHT CM ______

	CHOSEN PRODUCT	COL	MODEL 1 GM MCS NCS MP NP	SIZE	DROP	HEIGHT	NOTES and/or Alternative Code	OPTIONALS Button-hole	Travel bag
1									
2									
3									

REQUIRED DELIVERY DATE: | CONFIRMED DELIVERY DATE: | CUSTOMER SIGNATURE:

TESTING SUIT: (required changes) — JACKET MODEL SIZE DROP HEIGHT / TRAUSER MODEL SIZE DROP HEIGHT

CHANGES TO JACKET				CHANGES TO TROUSER			
G1	LENGTH (related to testing suit)	cm -	cm +	P1	LENGTH INSIDE LEG (related to testing suit)	cm -	cm +
G2	LENGTH SHOULDER - SLEEVE (related to testing suit)	cm -	cm +	P2	SHRINK BOTTOM / ENLARG BOTTOM	cm -	cm +
G3	WAIST SIZE	cm -	cm +	P3	WAIST SIZE	cm -	cm +
G4	UPPER LEG (HIP) SIZE	cm -	cm +	P4	CURVING	cm -	cm +
G5	SHOULDER WITDH	cm -	cm +	P5	CROTCH	cm -	cm +
G6	CANNELLI	cm -	cm +	P6	CROTCH	cm -	cm +
G7	LOW SHOULDERS DX☐ SX☐ / HIGH SHOULDERS DX☐ SX☐	cm -	cm +	P7	ASYMMETRIC FRONT CURVING / ASYMMETRIC BACK CURVING	cm -	cm -
G8	ADD CURVE (curved person) / REDUCE CURVE (backward person)	cm +	cm -	P8	UPPER LEG SIZE	cm -	cm +
G9	NECK GIRTH	cm -			CHANGES TO WAIST COAT		
G10	WIDEN CHEST	cm +		H1	LENGTH (related to testing suit)	cm -	cm +
G11	RESTRICT CHEST	cm -		H2	RESTRICT WAIST / WIDEN WAIST	cm -	cm +
G12	AROUND SLEEVE WIDTH	cm +		H3	ADD CURVE (curved person)	cm +	
G13	RESTRICT NAPE	cm -		H3	REDUCE CURVE (backward person)	cm -	

REPORTED MEASURES ARE FULLY SUBSCRIBED BY THE CUSTOMER UNDER HIS OWN RESPONSIBILITY. THE CLOTHING MADE CONFORMING THESE MEASURES IS A CUSTOMER PROPERTY.

Figure 3.3: Form for the customization of a suit

The form used for customization, however, is not a magic wand that solves the problems of configuration. As our expert says, ‘The information collected on the form cannot be supplied by the customer nor by an apprentice salesman.’ The support given by the shop assistant to customization is fundamental and, even though it is possible to select and hire ‘an interested and competent person’, ‘more than two years are needed to learn to personalize properly’. The paper form, which needs to be clearly understood by production, uses a language that is very different from the one used by the customer (‘it’s a bit tight here, it hangs a bit there’, etc.) and is a language that the salesman must learn. At the same time the shop assistant learns to understand ‘how the different parts of the garment are placed in relation to one another’, since the paper form is not able to provide a congruent verification of the specifications. In other words, the investment that must be made to train the salesman is not irrelevant and, what is worse, it does not belong to the shop: if the salesman moves on to a competitor, his knowledge moves on with him. Finally, even in the case of expert shop assistants, it is possible that production will contact the shop because it has found differences in the specified information. This causes problems of delivery delays and imperfect customization, as we have already mentioned.

3.3 Structured approaches to collecting and checking specifications

When the product is characterized by a certain level of complexity, when possible configuration mistakes become difficult to correct and when the interdependencies among customizable attributes tend to become numerous and articulate, then even approaches directed to structured collection of specifications show evident gaps. They do not provide real support when checking compatibility and validity of product specifications. Therefore, it is necessary to use some instruments that can help the salesman not only to collect information in a structured way but also to check the correctness of this information. Among the most common instruments of this type we find: double entry tables, double entry diagrams and multiple entry tables.

Double entry table The double entry tables enable one to single out, starting from only two parameters, the desired configuration or a set of product variants that satisfy the conditions established by the two parameters. The parameters are laid out along lines and columns, so that the information searched can be found in the cell that results from their intersection. The nature of such information can vary according to the purpose of the table and the characteristics of the product.

One possibility offered by the double entry table is to specify a product code. Given two parameters, one for a line and one for a column, it is possible to find in their intersection the product code or a sign that indicates that the product exists or is feasible.

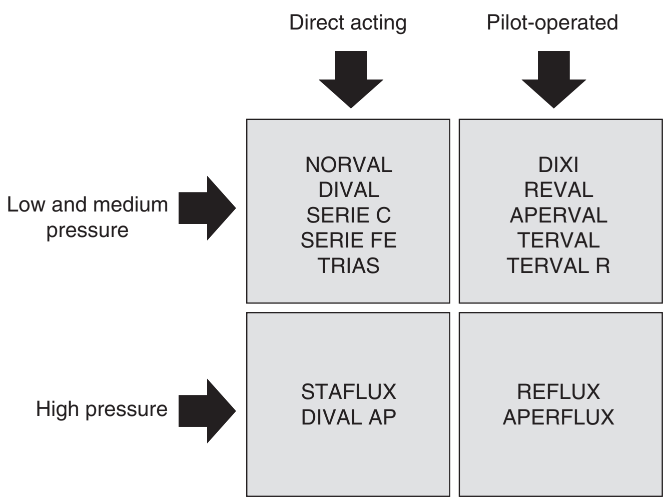


Figure 3.4: Double entry table for finding product families (gas pressure regulators)

Alternatively, the double entry table can help one to select a specific product family that satisfies certain requirements. Pietro Fiorentini Spa, a well-known producer of systems and components for gas regulation, for example, uses a double entry table to guide customers towards the product family that best fits their needs. This table distinguishes the regulator family according to (1) the working pressure and (2) the way the regulation device works (see Figure 3.4).

A third purpose of the double entry table is to support the verification of compatibility among the choices made by the customer. In this case, the cell shows a conventional sign, e.g. a tick (✓) to indicate the compatibility of the two alternatives. A company that produces air conditioners, for example, has a series of accessories for its split machines – rods for installing the external unit, cooling liquid insulation lines, devices for usage at low temperatures, etc. – which are produced in different variants according to the type of conditioner that has been selected (see Table 3.3).

Double entry diagrams Double entry diagrams derive from the documents used in the technical office to identify the relationships between two parameters. To understand how these diagrams work, we can consider the case of a company that manufactures volumetric pumps. A pump is characterized by two fundamental parameters: the fluid capacity that it can process and the head, i.e. the fluid pressure in the pump output. Each pump model works efficiently only in a set field of pres-

Table 3.3: Double entry table for checking the compatibility of the choices made by the customer

	Compatible accessory									
Mod. CR	*31*	*51*	*71*	*101*	*151*	*31H*	*51H*	*71H*	*101H*	*151H*
DR 33	✓									
DR 53		✓								
DR 78			✓							
DR 103				✓						
DR 153					✓					
FRC 31	✓					✓				
FRC 51		✓					✓			
FRC 76			✓					✓		
FRC 101				✓					✓	
FRC 151					✓					✓
PGS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

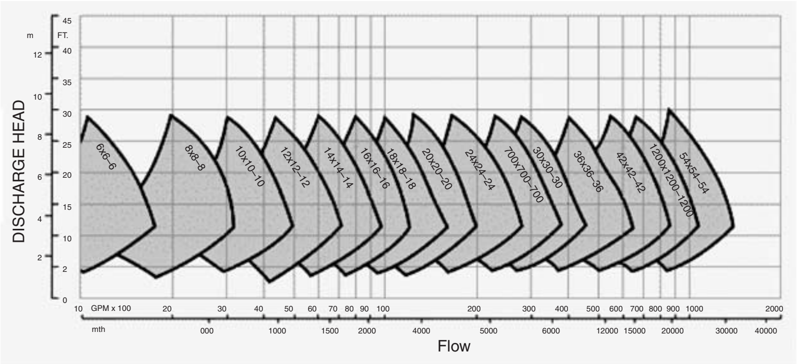


Figure 3.5: Double entry diagram for a volumetric pump

sures and heads. The problem consists in finding the type of pump that best meets the customer's requirements. Figure 3.5 shows a double entry diagram that describes the usage fields (in terms of capacity and head) of different pump models.

As seen from the figure, the main difference between a double entry diagram and a double entry table is that in the former, it is possible to establish complex relationships between product attributes. If we observe the shape of the working fields in the various models, we can see that the same pump can develop a capacity of 400 to 700 m^3/h, supplying a 6 metre water column head, but the capacity field narrows when the head becomes 8 metres c.a. A double entry table would be suitable for representing the working field of a pump if pressure and head were independent parameters, i.e. if the XYZ pump could independently elaborate, for example, capacities from 500 to 600 m^3/h with a head from 7 to 9 metres c.a.

Multiple entry table Should it be necessary to highlight the multiple parameters that characterize a certain product contemporaneously, the most suitable instrument is a multiple entry table. Typically, these tables show the product variants in the columns (lines) and the characteristics that make these variants different along the lines (columns). Let us consider, for example, a cooling machine (chiller) the main component of many conditioning systems. Table 3.4 shows a series of product variants made by one of the main national producers, in the columns (151, 201, etc.), in order of increasing power.

A series of technical characteristics grouped by models (WRAT, WRAD, etc.) are indicated along the lines. The multiple entry table has

Table 3.4: Multiple entry table

MODELS		*0151*	*0201*	*0251*	*0301*	*0302*
WRH/WRHH						
Refigeration only						
Cooling capacity	kW	43	48	61	71	86
Power input	kW	11	12	16	19	21
Condenser heating cap.	kW	53	59	76	89	106
Heating only						
Heating capacity	kW	51	57	73	85	101
Power input	kW	13	14	19	22	25
WRHD						
Cooling capacity	kW	45	49	63	73	89
Power input	kW	10	12	16	19	21
Desuperheater heating cap.	kW	9	10	13	16	18
WRHR						
Cooling capacity	kW	39	43	55	64	78
Power input	kW	13	14	19	22	25
Heat recovery cap.	kW	51	57	73	85	101

the advantage of communicating the range of variants offered by the company 'at a glance': the columns placed under each model provide a set of characteristics for each product variant. Moreover, the table makes it possible to follow different ways to reach the identification of a product variant. For example, if the conditioning system designer is working on a new building, the first feature to consider for the configuration will be the cooling power, and afterwards the other parameters will be verified. On the other hand, if the designer has to install the system into an already existing building, then parameters such as machine dimensions and weight will become the first features to be considered, dictated by the machine installation area and the floor dimensions.

Pros and cons of tables and diagrams As seen from the previous examples, tables and diagrams have a series of advantages when obtaining and checking product specifications (see Table 3.5).

Firstly, when using tables and diagrams the product offer is communicated more clearly than when using pre-printed forms. Unlike pre-printed forms, tables and diagrams not only show the product attributes, about which the customer can express a preference, but also communicate the range of available alternatives for each attribute.

Secondly, they often support the process of identification of the product variant that satisfies the customer's needs. They offer the possibility of singling out, first the product family (Figure 3.4) and then the appropriate variant within that family (Table 3.4) that meets the customer's requirements.

A third advantage of tables and diagrams is that they make it possible to check compatibility ties between the options chosen by the customer – something that is clearly not possible with a pre-printed form.

Tables and diagrams, however, have some important disadvantages (see Table 3.5). Firstly, the commercial office staff and in some ways also the customer, need a relatively high technical knowledge to use these configuration instruments. In fact, it is necessary to understand the meaning of the specifications shown in tables and diagrams as well as the interdependence among them. Moreover, it is also necessary to consider 'facts that are not written in the official documents'. For example, when dimensioning a pressure regulator, it is possible to single out two alternative product variants that create the requested pressure jump, one larger and one smaller. The larger one will be more suitable if the customer forecasts an increase of gas consumption, since he will have to process a greater gas flow. In contrast, if the customer's consumption is reasonably stable and cost is an important sales factor, the smaller regulator will be preferable. The difficulty in using this kind of document makes it necessary to have an expert, and therefore a costly sales force. Such a sales force becomes indispensable because it is very difficult for the customer to use the complex documentation on his own to find the correct product configuration.

Table 3.5: Advantages and disadvantages of tables and diagrams for product configuration

Advantages	*Disadvantages*
✓ Good communication of company's product assortment	✗ Good technical expertise is needed to use charts and tables
✓ Fast identification of product variant needed by customer	✗ Difficult and expensive to maintain and update technical documentation supporting sales
✓ Simple constraints between specifications can be expressed	✗ Difficult to give the customer reliable information on product price and delivery data

The second problem is represented by the investment necessary for elaborating, printing and updating this documentation. When the company has dozens of models and thousands of product variants, the cost rises very rapidly. Furthermore, the more the product variants, the more frequent the modifications and creation of new models. The result is a great quantity of obsolete documentation: piles of forms are directly destroyed and the last printed versions do not reflect what the company can really offer. The company has to invest large amounts of money and finally it only gives the customer and the salesman a support that is complicated to use and often not completely up to date.

Tables and diagrams used as supports for product configuration do not solve the problem of price and delivery time determination. These two parameters are fundamental in the order acquisition process and very often depend on the product configuration already identified.

3.4 Dynamic approaches to product configuration

All the approaches to product configuration described up to this point share an important aspect: they support the commercial configuration but not the technical one (see case 2.1). It does not matter whether a company uses loose sheets, pre-printed forms, diagrams or other instruments – once the specifications have been collected, the problem of singling out or generating the technical documentation necessary to manufacture the product, still remains. It is a complex problem, due to the large quantity of technical information that is often needed.

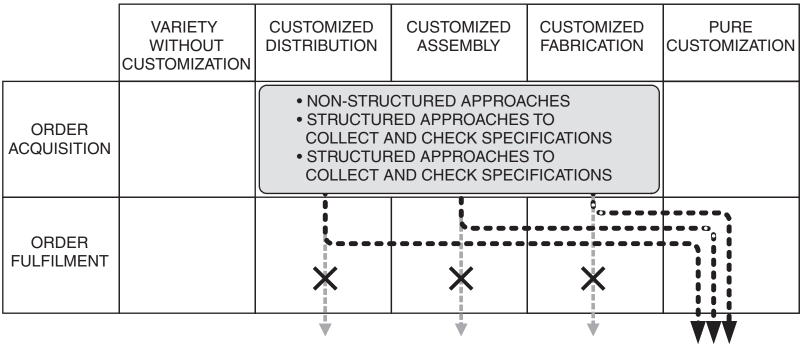


Figure 3.6: Pure customization due to lack of support for the technical configuration process

The consequence of this lack of technical documentation is that the company, in the end, uses pure customization in the product configuration process, even if this is not logically necessary (see Figure 3.6).

The first software programs created to remedy this lack appeared in the 1980s: these tools were designed for a particular type of product and for this reason they are known as *special purpose configuration software*. Even in this case we will refer to a practical example to understand the positive and negative aspects of this kind of solution.

Case 3.3 Configuring a microcomputer: the digital case

Digital Equipment Corporation (DEC), later taken over by Compaq and then by Hewlett Packard, was one of the pioneers in the information technology sector. This company invented a market in which it remained the unchallenged leader throughout the 1970s and 1980s: the microcomputer market. Even though less complex than the more costly mainframes produced by IBM, Honeywell, Bull, etc., minicomputers were highly configurable and towards 1980 the number of possible configurations was around one million, making it impossible for a single person to handle manually all the information necessary to carry out a correct configuration. In spite of the fact that specialized salesmen and technical editors who were responsible for controlling configuration correctness were employed, a survey showed that in 35 per cent of the cases there were configuration errors, and the average time for the complete configuration cycle was of one to two days, even for relatively simple machines.

To tackle these problems, DEC decided to develop an ad hoc (special purpose) configuration software made up of separately designed modules that were subsequently integrated. The first module was called X-SEL and guided the salesman towards the identification of customer needs in order to find the best product configuration. The second module was called X-CON and was positioned at back-office level: it dealt with the technical configuration, i.e. the elaboration of the product specifications to obtain the necessary documents for computer production and assembly (see Figure 3.7).

By using this system, the number of configurations carried out each year by the sales office personnel increased considerably, jumping from an average of 650 configurations for each salesman in 1981, to nearly 1300 in 1986. During the same interval, the correctness of the configurations collected increased from 65 per cent to 98 per cent. The average time requested for order fulfilment, finally, was reduced from three months to little more than three weeks.

The advantages are evident but there are still some difficulties associated with the implementation and subsequent use of the system and with the organizational changes (see Table 3.6) The first difficulty, typical of any special purpose system, concerns the high cost associated with the development of the software tool, a cost that makes this solution accessible only to large companies. In the past, many companies attracted by the solutions offered by software houses tried the

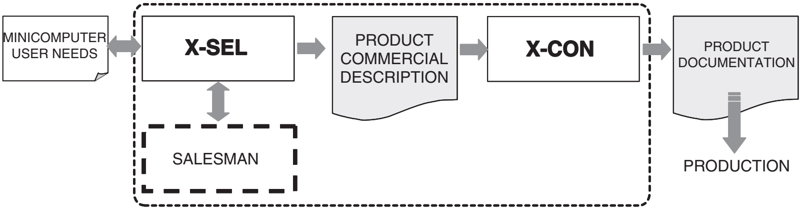


Figure 3.7: DEC configuration system

special purpose configurator (i.e. specially developed according to personal needs). The considerable investment and the complexity of the programs, however, caused the failure or abandonment of the project.

The second problem is the management of *product knowledge*. To understand this, we have to recall that the DEC configurator represented the product using a set of IF … THEN statements, i.e. it was a *rule-based* product configurator. Such an approach usually leads to complex and difficult-to-update product models. Given minicomputers' inner complexity and fast-paced evolution, DEC incurred serious difficulties in keeping the product configurator updated. Incredibly, DEC had to use between twenty-five and fifty-nine employees in 1989 to update around 40 per cent of the product data stored within the configurator.

The third serious problem was due to the fact that since a special purpose configurator has a clearly determined use, the construction of an interface supporting product modelling cannot be justified. For this reason the system is maintained and modified by programmers rather than by its users – technicians and salesmen. A gap is therefore created between users and developers of the system. At this point making the system effective becomes more important than simplifying its use, limiting the application of the software program in the company.

Table 3.6: Advantages and disadvantages of special purpose configuration software

Advantages	*Disadvantages*
✓ Highly correct configurations	✗ Very high initial investment to set up the system
✓ High salesman productivity	✗ Difficult to maintain
✓ Reduced order fulfilment time	✗ Personnel may oppose implementation

3.5 Beyond traditional approaches

The problem of product configuration is not new, even if nowadays it is becoming more relevant and extended. In order to tackle this problem many approaches were developed in the past, each one with its own advantages and disadvantages. However, none of the traditional approaches completely satisfies a company that has to elaborate a large number of product configurations with at least a minimum of complexity. In the mid-1990s, a series of new conditions contributed to the development of modern configuration systems:

- the growing capacity of computers;
- the development of programming techniques directed to powerful and flexible objects;
- the advanced studies of artificial intelligence on the 'constraint satisfaction problem';
- the growing knowledge about information technology applied to product configuration in many large innovative companies.

The modern product configuration systems that will be fully described in the second part of this book combine many of the advantages of the traditional approaches and, at the same time, present fewer disadvantages.

This page intentionally left blank

Part II

Product Configuration Systems

This page intentionally left blank

4 Configuration Systems Architecture

When the product configuration tasks increase in complexity, traditional approaches show serious deficiencies. This chapter illustrates how modern configuration systems overcome the difficulties of product configuration. In particular the chapter:

- describes modern configuration software architecture, pointing out its fundamental components and functions;
- illustrates the concept of a configuration system, considered as a set of technical and human resources;
- discusses different alternatives for the definition of a configuration system;
- delimits the domain where product configurators are applicable, distinguishing them from selectors and meta-configurators.

4.1 Product configurators architecture

A company that offers customization via configurable products is able dynamically to adapt its products to a wide range of customer requirements, without carrying out design activities. This approach differs from pure customization, where the product variants are designed 'to order', and from variety without customization, where operational processes are not tied to order acquisition processes. The possibility of 'customizing without design' is connected to the concept of 'product configurability', i.e. the possibility of offering a complete description of product family architecture. This means that:

- the company has defined 'a priori' all the possible variants of all the possible components used in a product family, and/or all the 'rules' according to which the components variants are determined;
- product functions and purposes – and therefore their characteristics – have been associated with single component variants or with their combinations.

The first principle establishes that the elaboration of any product variant does not require new components – or if it actually needs some, the description of such components is obtained by using algorithm instructions. The second principle establishes that it is possible to associate, without uncertainties, the functions and purposes required by the customer with a given set of components. The final result, when defining the architecture of a product family, consists in the fact that the company has virtually eliminated any kind of uncertainty about the product characteristics. But, in the configuration process, this elimination is not enough to guarantee a fast and correct output. Configuration tasks, in fact, are characterized by little or no uncertainties and they may also be characterized by high complexity. This complexity in the configuration tasks derives from the fact that a great deal of information must be processed: checks, translation of customer needs into product commercial characteristics and vice versa, verifications of completeness and congruence of the specified commercial characteristics, etc. On the one hand, the definition of product architecture makes the configuration process logically possible, without involving the technical office. On the other hand, it is also necessary to define instruments to support configuration that can reduce computing complexity, giving the salesman – and even the customer – the conditions to operate autonomously in a fast and correct configuration process. From this point of view, let us consider again the case of a suit. The form 'reminds' the shop assistant of all the parameters that are specified in the customization of the product. Diagrams and double entry tables give some simple information to the salesman, who has to specify the values of the parameters written on tables and diagrams and point out the information associated with their intersections.

The structured approaches described in the previous chapter can be considered as attempts to reach a formal, even if partial, description of product architecture. For example, the form used for personalizing a suit implicitly describes all the parameters necessary to adapt the garment to a specific customer. This form supplies a commercial description of the suit. Diagrams and double entry tables, in contrast, supply a general description mainly focused on technical aspects, pointing out product components and links between commercial characteristics. It is obvious that these instruments cannot support the 'whole' configuration process, as seen in section 3.4. Therefore, it is necessary to carry out some design activities in the order fulfilment process. As a consequence, the companies cannot offer customization through configurable products in a pure way.

A complete description of the product, from the commercial as well as from the technical point of view, is obtained by using 'special purpose' configurators. If we consider, for example, the operation of X-SEL (case 3.4) we can see that it had a double function: (1) it described the specifications that could be determined by the client, and (2) it guided the salesman in the process of collecting information. In other words, the system incorporated part of the company knowledge about the commercial description of the product and the customer requirements. A software system capable of supporting the commercial configuration of a product requires a definition of the commercial model.

> COMMERCIAL MODEL: *a formal representation of the product space and of the procedures according to which a commercial configuration can be defined within such space.*

The configuration system created by DEC incorporated part of the company knowledge about the technical description of the product used for translating the commercial specifications into product documentation. A software system capable of supporting the technical configuration of the product requires a definition of the technical model.

> TECHNICAL MODEL: *a formal representation of the links between commercial characteristics and the documents that describe each product variant (bills of material, production and assembly cycles, etc.).*

Once the commercial model and the technical model have been developed, they must be interpreted by suitable programs known as configuration engines. The *commercial configuration engine* reads the data stored in the commercial model and interacts with the user in the commercial configuration process. Similarly, the *technical configuration engine* reads the data stored in the technical model and supports the engineering and manufacturing activities needed to deliver the product (see Figure 4.1). It is important to notice that the scheme supplied is a

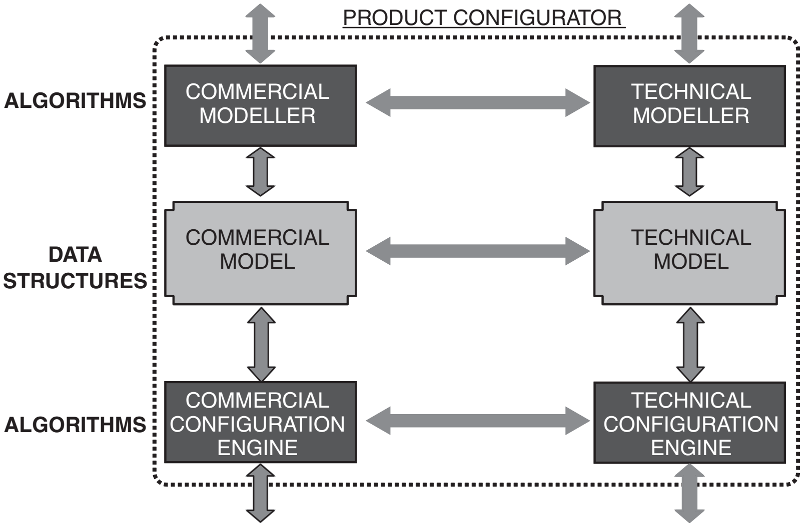


Figure 4.1: Logical architecture of a product configurator

logical one, meant to single out the fundamental activities of the product configurator. The software architecture in a real configuration system can be more or less similar to the one described here. For example, the two configuration engines are not always two separate components of the program. However, even if only one module of the system performed the commercial as well as the technical configuration, it would carry out two different functions.

The need to build a commercial model, as well as a technical one, in order to reduce computing and knowledge complexity in configuration tasks, raises the question of how to build such models. In the case of X-CON–X-SEL, the product model was mainly obtained by writing a series of conditional statements IF … THEN, and its updating required deep knowledge and competence. The difficulty found in the modification of the commercial and technical models involved two serious problems. Firstly, the normal users of the system, even the most expert ones, did not have any chance to modify such models, even in the case of very simple changes. Users had only an indirect control over the configuration process supported by the computing system. Secondly, the modification of the commercial and technical models presented a complexity that obstructed the updating of the system, determining a gap, or at least a delay, between new manufactured product variants and those controlled by the system. The product configurator can be effectively adopted by the company, i.e. largely and regularly used, if it includes suitable instruments that facilitate the creation and updating of the commercial and technical models.

> DIALOGUE MODELLER: *a computing tool that supports the creation and updating of the commercial model.*

> PRODUCT MODELLER: *a computing tool that supports the creation and updating of the technical model.*

The software used for modellers has the same characteristics as that used for the configuration engines. It is not necessary to have two separate modules, one for the dialogue modeller and one for the product modeller. However, the software will perform two different modelling functions. Figure 4.1 shows the logical architecture of a product configurator. As seen in the figure, a fundamental characteristic of modern product configurators is the fact that they include advanced modellers that are flexible and can be operated by users who are not expert programmers, and can be used to represent the most varied range of products. In fact, the user will create, for each particular case, the specific technical model and the specific commercial model that best suit his product, utilizing these modelling instruments.

4.2 From configurator to configuration system

To think that a configuration system is something exclusively limited to a computer tool – the product configurator – would certainly be a restrictive assumption. It is obvious that, during the configuration process, the software interacts with the personnel at least in the initial and final stages. As seen in the previous section, the software also interacts with the personnel during the modelling process.

> MODELLING PROCESS: *the set of activities through which the necessary knowledge for product configuration is gathered and processed into commercial and technical models.*

Therefore, the computing system is linked with two processes – the configuration process and the modelling one – by means of two interfaces: the configuration engines and the modellers. Taking into account these notions, we can define the concept of a configuration system.

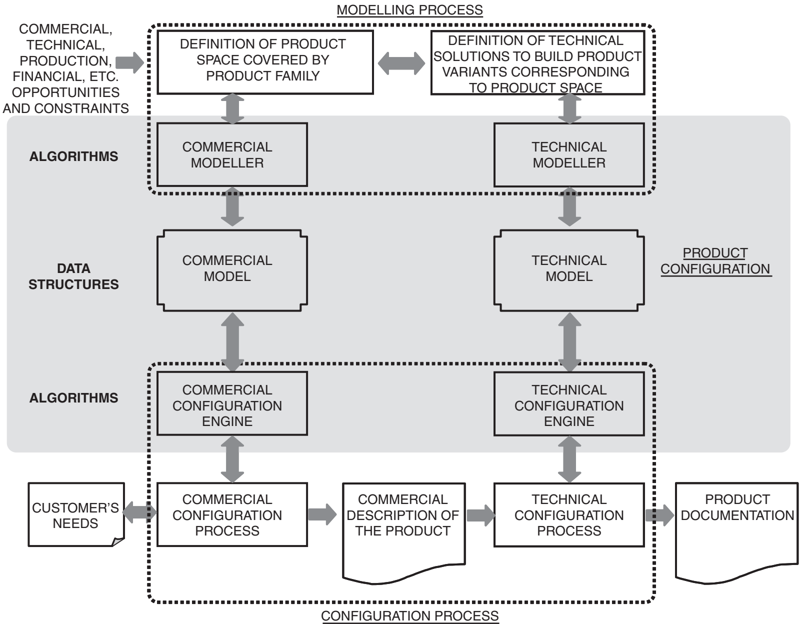


Figure 4.2: Logical architecture of a product configuration system

> CONFIGURATION SYSTEM: *the set of human and computing resources that contribute to accomplishing the configuration and modelling processes.*

The schematic representation of the essential components of a configuration system is shown in Figure 4.2.

The configuration system consists of a 'technical' dimension, made up of machines, product models, programs, etc., and a 'social' dimension, made up of the formal and informal organization of people and by the rules that, within the company, support the configuration process. A configuration system can be defined as a *socio-technical system*, whose optimization requires the combined optimization of the human and computing sub-systems.

The integration of both sub-systems depends on the conditions under which the system is introduced, that is to say, on the specific constraints imposed by the operational context where the configuration tasks are carried out. Product and commercial transaction complexities, the company's capacity to invest, the production volume, the number of configurations, and the type of customer (consumer or industrial) all contribute to determine the way in which the human and technical components of the system are integrated. The socio-technical system should be considered as an open and dynamic system in relation to the external environment. In fact, the external environment, through a series of contingent links, helps to define the most suitable form of the system that will be adopted.

The next three sections describe three fundamental alternatives for building configuration systems, as well as the conditions that determine the appropriateness of each alternative. These three alternatives are differentiated by the level of automation used during the configuration process, which in a certain way shows the importance of the technical and human components of the system.

4.3 Moderately automated configuration systems

A configuration system with weak automation is a system that does not entirely replace the human operator in any of the logical moments of the configuration process. The system supports the operators within the commercial and the technical office, respectively, in the commercial and technical configuration activities. The configuration process is carried out in two successive stages; in each of them the computer system interacts with the human operator in order to gather the decisions taken in relation to the commercial and technical configuration. Therefore, there is a *configuration in two successive stages*; each stage specifies only the product characteristics associated with its competence (see Figure 4.3).

In the first stage, the characteristics perceived by the customer about the product functionality are defined, without taking into account design or production aspects and without asking for the necessary

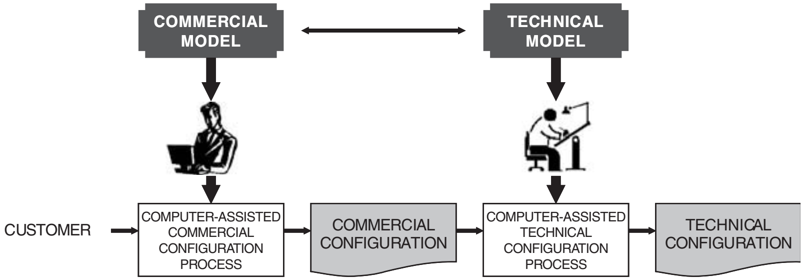


Figure 4.3: Configuration system with weak automation

information to elaborate the bill of material. The output of this first stage is the commercial configuration that is used, in the second stage, by the technical office to plan production activities. The documents needed to manufacture the product, however, require a second 'dialogue' to define the technical parameters of the product. Typically, the last word in this dialogue is not from the customer, but from a technician. He is responsible for defining the technical solutions that are compatible, not only the with commercial specifications, but also with the economic, technological, production and supplying aspects.

Case 4.1 Configuration with weak automation for the textile industry

Penelope Inc. operates in the international market of textile looms, where it occupies a leading position. The sales force is made up of salesmen who directly visit the customers in their companies to promote and sell its products. Once the deal is about to be sealed, that is to say, when the salesman has discussed the commercial terms of the sales contract with the customer and has defined the technical characteristics, he sends a fax to the commercial office including all the information gathered.

At this point, an employee from sales support receives the fax and elaborates the real and proper sales order. Automatically, the configuration process starts. The information sent by the salesman – about 100 different independent specifications – is entered into the system. The system then specifies and evaluates the further dependent characteristics – which are about 100 – provided by sales support. During this first part of the configuration process, focused on the commercial specifications of the product, it is not possible to include all the data, in particular those properly and specifically technical, needed for the complete configuration of the loom. Even if sales support is not able to specify all the characteristics, it may answer some questions by giving provisional answers (maybe using default values and the descriptive text related to each question) or may give no answer at all, because in the product models of Penelope there are no obligatory characteristics/questions, in order to create a flexible configuration that can be managed by more than one person. Once all the questions of the configuration dialogue have been analysed, sales support adds some notes (written using R/3 system or MS Word) to a list of questions, with possible explanations, that were not answered, as well as the characteristics that need correction or further specifications.

At the end of this first step, the sales order is sent to an employee in the production area, who carries out the second part of the configuration process, properly technical, and the complete selection and evaluation of all the independent characteristics is made. This employee reads the notes sent by sales support and concludes the configuration introducing all the necessary data to elaborate the software. If the employee is in doubt about a particular specification, he may ask for the collaboration and help of an expert from the technical office, who knows the product perfectly, since he does not follow the planning activities and only participates in assisted configuration when his collaboration is needed.

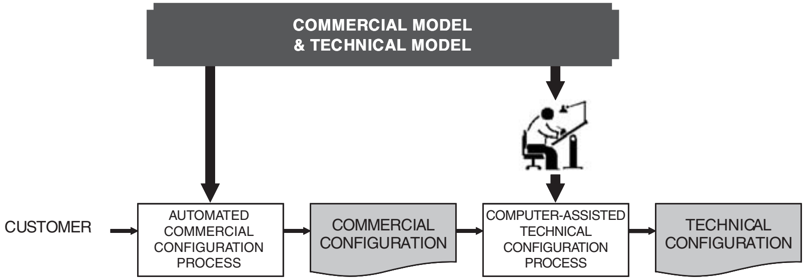


Figure 4.4: Highly automated configuration system (fully automated commercial configuration)

4.4 Highly automated configuration systems

When the product configurator partially supports one of the configuration activities (commercial or technical) and the other one is completely automated, we are dealing with a *configuration system with strong automation*. The personnel take part in the commercial configuration (the technical activities are fully automated) or in the technical configuration (the commercial activities are fully automated)

In the case of *fully automated commercial configuration* (see Figure 4.4) the technical office plays an 'active' role in the configuration process, regularly influencing its output.

This case may occur when a company decides to offer a certain product on-line, but investments and production times do not allow the company to build a product model with a fully automated technical configuration. The rapid elaboration of a satisfactory commercial model is subordinated to a predetermined rationalization of the line of products. In other words, the company must conceptualize 'a priori' the line of products, defining the attributes the customer can select as well as the available choices for each attribute. This kind of system is appropriate when relatively simple combinations of commercial specifications are needed, although they may require a complex planning effort. But if, during a long period of time, the demand for configurations is very low, the investments made for the construction of a complex product model may not be profitable.

Case 4.2 Configuration with strong automation of voltage transformers

Blackout Ltd. is a small company that produces customized transformers. Adopting a customization strategy, Blackout Ltd. produced 'out of catalogue' more than 70 per cent of the transformers sold in around six years, for a total of more than 10.000 different variants. Under these circumstances and due to the pressures of costs and delivery times (see case 3.1), Blackout Ltd. decided to introduce a configuration system. In the case of more complex product lines, only the commercial configuration was completely automated. In fact, the most complex products are not completely different from the simplest ones, in the definition of their commercial characteristics, but it becomes difficult to automate their technical configuration.

The configurator does not replace the designer; on the contrary, it supports him in a second configuration dialogue, intended to gather the remaining designing parameters that must be defined. The designer, using appropriate calculations and his own experience, determines and incorporates the values of this second dialogue. The advantages for the designer are: (1) he immediately has a precise idea of what kind of configuration the customer wants, and (2) he knows which characteristics must be defined (this is the field of the second configuration dialogue). Furthermore, the system is able to check the completeness and validity of the data introduced by the designer, ensuring a further control in the product configuration process. Once all the technical characteristics have been specified, the system 'translates' the information provided by the designer into bills of material and production cycles.

In the case of *fully automated technical configuration* (Figure 4.5), only the commercial office plays an active role in the configuration process. This situation occurs when the commercial configuration requires particular attention, for example, if the product has a high cost or if there are difficulties in the installation or if there are emotional factors that must be managed by the salesman. Let us consider, for example, the case of automobiles sold on-line. Fiat, in spite of the commercial success of the sports car Barchetta, was only able to sell seven automobiles on-line. Having a 'live' approach to the product, discussing and bargaining with the salesman and listening to possible advice are all fundamental aspects that influence the customer's decision.

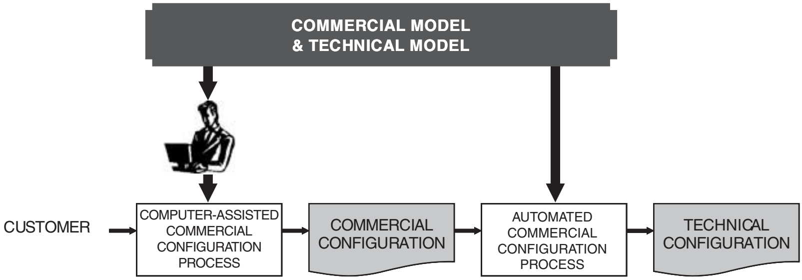


Figure 4.5: Highly automated configuration system (fully automated technical configuration)

Fully automated technical configuration is possible even in the case of complex products, when their structure has been simplified in order to make the configuration process relatively flexible: for example, in the case of highly modular products such as telephone exchanges, mainframes, etc. A simplification of the product architecture, and consequently of the product model and its manufacturing, does not necessarily mean a simplification in the definition of commercial specifications. In fact, in the presence of numerous commercial characteristics, it is difficult to interpret all of them and understand the capability that alternative product configurations may have in satisfying customer requirements.

Case 4.3 Configuration with strong automation for numerical control milling tools

Grooveperfect is a leading company in the production of high speed milling machines. Like many other companies in this sector, as a result of business development and expansion, it started feeling the 'quasi unconscious' need to adopt structured approaches for product configuration. It became necessary to redesign the machines, that are made up of thousands of fundamental components, in order to simplify their characteristics, technical manuals and spare parts lists, which differ according to each variant. The rationalization of the product structure made it possible to build, for each machine family, models that can automate the technical configuration. The commercial configuration, however, is semi-automated, since there is an 'ad hoc' function called *sales engineering*, that interacts with the customer in the collection of specifications and introduces them in the configurator, thus feeding the technical configuration process. The staff of the sales engineering department is made up of people who are experts in milling machine applications rather than in their manufacture. Therefore, they first analyse the customer's requirements and then find a solution, based on their knowledge of different mechanical production aspects, of the product itself and of the accessory and tool suppliers. In synthesis, the staff of the sales engineering department interprets a need that the customer expresses, but that he is not always able to 'translate' into a coherent set of product specifications – due to the intrinsic product complexity or to an inadequate understanding of the product, above all in the case of small companies.

4.5 Totally automated configuration systems

When the configurator carries out all the configuration activities, commercial as well as technical, without involving the company personnel, we have a *totally automated configuration system* (see Figure 4.6). Evidently, this type of system accelerates the responses given to the customer and considerably reduces the participation of the commercial and technical staff in repetitive activities or transactions related to

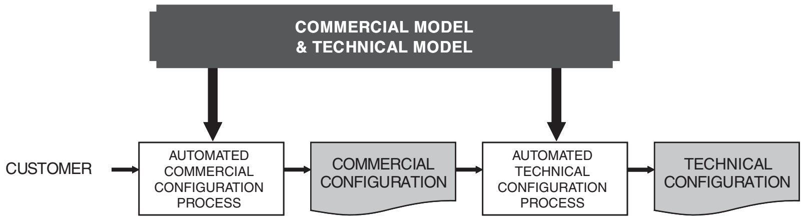


Figure 4.6: Totally automated configuration system

product customization (dimensioning, controls, product documents modifications, etc.). For this reason, configuration systems with total automation became a fundamental aspect in the information infrastructure necessary for 'e-business' in companies that offer customized products.

A system that automatically carries out the entire commercial and technical configuration must be considered carefully, because it lacks the typical redundancy of human control and because it may imply very high investments.

This system is appropriate when the customer is able to define, autonomously, the commercial characteristics, and when the product model is not excessively complex. The 'famous' examples of this system are the on-line configurators for personal computers and notebooks, of which Dell (www.dell.com) was a pioneer.

Case 4.4 Configuration with total automation for hydro-massage bathtubs

Wholly Bubble is a fast growing player in the industry of bathtubs and showerboxes with hydro-massage. The high competitiveness of the market urged the company to offer the possibility of adapting its products to different requirements. The countless combinations of tubs with control panels, water jets, accessories, chromo-therapy, music-therapy, etc. enabled the company to offer thousands of different product variants. The fact that, for aesthetic, functional, technical and safety reasons, the different accessories could be incompatible increased the number of configuration errors, with consequent problems of installation, safety and image. For this reason, Wholly Bubble implemented a configuration system. At the beginning this system only automated the technical configuration, that was not an excessively complex task, due to the type of products. The company took advantage of the particular way in which the products are distributed. Bathroom fittings are mainly distributed by concessionaires run by experts in the sector. The interface with a mature customer allowed the company to operate without the participation of the personnel, including the collection of the commercial characteristics. In the end, by automating both sales and technical configuration tools, Wholly Bubble ended up creating a system with total automation.

4.6 How to determine the optimal degree of automation

Taking into account the previous discussion, it is possible to identify a few key variables affecting the degree of optimal automation of a configuration system. More specifically it is crucial to take into consideration at least the following:

- product complexity;
- number of configurations made in the time unit;
- degree of product knowledge on the part of the customer;
- time and resources available for the creation of the configuration system.

Figure 4.7 shows a qualitative description of how the degree of optimal automation in a configuration system changes with the variation of these four parameters.

4.7 Configurators, selectors and meta-configurators

We speak of configuration when the customer requirements directly influence the distribution, assembly and fabrication of the product, but

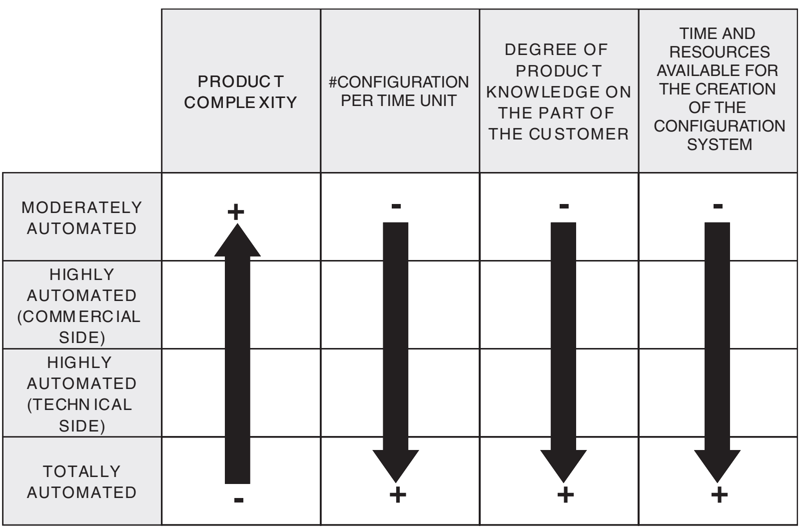


Figure 4.7: Factors that influence the degree of optimal automation in a configuration system

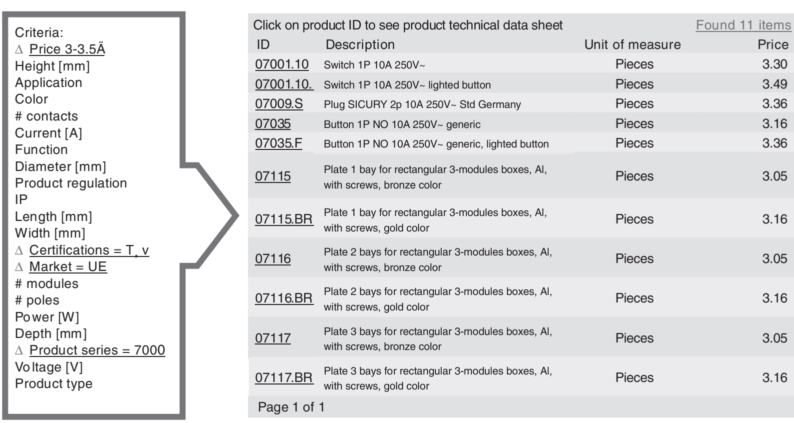
not the planning activities. As previously discussed, a key problem in product configuration is the selection of options and product attributes. A similar problem appears even in the case of pure customization or in certain cases of variety without customization. In these contexts, two specific types of information systems are used: the *product selectors* and the *meta-configurators*. They are similar – in some aspects – to product configurators, but they contain distinctive characteristics.

In the case of variety without customization, when many product variants are offered, a company needs to help its customers in identifying the variants that best suit their needs. Such help is aimed at reducing the costs customers incur in singling out the product variant that best fits their needs. The solution in this case can be a *product selector*: an instrument that can help the client to scan through the product variants offered by the company, thus supporting his selection process.

Case 4.5 Product selector for low voltage electrical fittings

Vimar, a company that produces low voltage accessories and equipment for control, protection and distribution of electrical power in the private and service sectors, created an example of a product selector. The thousands of different variants and models offered, urged the company to give the customers and installers an opportunity to consult an 'intelligent catalogue' on its web site, i.e. a product selector (see Figure 4.8).

The mechanism of the selector is simple and immediately understood: the customer/installer specifies a series of parameters that define the product he is looking for (e.g. products of the series 7000, at a low price between €1.50 and €1.75, with IMQ standards and that can be sold in the European Union



ID	Description	Unit of measure	Price
07001.10	Switch 1P 10A 250V~	Pieces	3.30
07001.10.	Switch 1P 10A 250V~ lighted button	Pieces	3.49
07009.S	Plug SICURY 2p 10A 250V~ Std Germany	Pieces	3.36
07035	Button 1P NO 10A 250V~ generic	Pieces	3.16
07035.F	Button 1P NO 10A 250V~ generic, lighted button	Pieces	3.36
07115	Plate 1 bay for rectangular 3-modules boxes, Al, with screws, bronze color	Pieces	3.05
07115.BR	Plate 1 bay for rectangular 3-modules boxes, Al, with screws, gold color	Pieces	3.16
07116	Plate 2 bays for rectangular 3-modules boxes, Al, with screws, bronze color	Pieces	3.05
07116.BR	Plate 2 bays for rectangular 3-modules boxes, Al, with screws, gold color	Pieces	3.16
07117	Plate 3 bays for rectangular 3-modules boxes, Al, with screws, bronze color	Pieces	3.05
07117.BR	Plate 3 bays for rectangular 3-modules boxes, Al, with screws, gold color	Pieces	3.16

Figure 4.8: Product selector for low voltage electrical fittings (see www.vimar.it)

market). The specified parameters generate a query that automatically takes information from the product database, showing the variants that suit the characteristics selected. At this point the customer/installer can get further information (installation, technical features, photos, etc.) regarding the variants that interest him.

In the case of pure configuration, the equivalent of attribute selection is performed by the designer, and not by the customer. The customer, in fact, does not have to look for any variant, because there is no catalogue. The problem concerns the designer, as he must rapidly evaluate what is the best, most economical, fastest and most suitable solution, from the technical point of view, to satisfy customer requirements. The designer will rapidly have to come up with a general product design that, even if approximate, must be reliable. The necessity of a support for this preliminary design, not for the detailed definition of all the design parameters, determined the creation of a type of information system that is called a *product meta-configurator*.

Case 4.6 Meta-configurator of prefabricated bridges

An example of a meta-configurator is used in the planning of prefabricated bridges. In this case (see Figure 4.9), the designer has to rapidly estimate the costs related to an archway (300m) over a riverbed characterized by a well-determined profile and by a certain maximum depth (70m).

The rules are grouped according to design criteria, such as technical regulations, safety standards, aesthetic features, economic aspects, etc. The order of these criteria is established by the user. If different rules conflict with each other during the meta-configuration task, the system may activate ad hoc conflict-

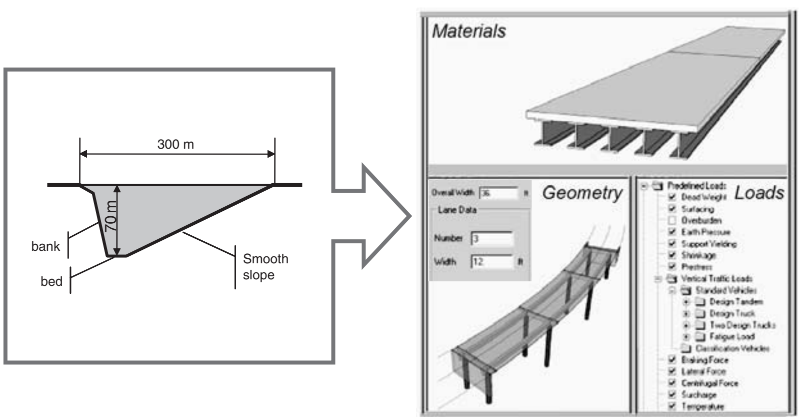


Figure 4.9: Meta-configurator of bridges

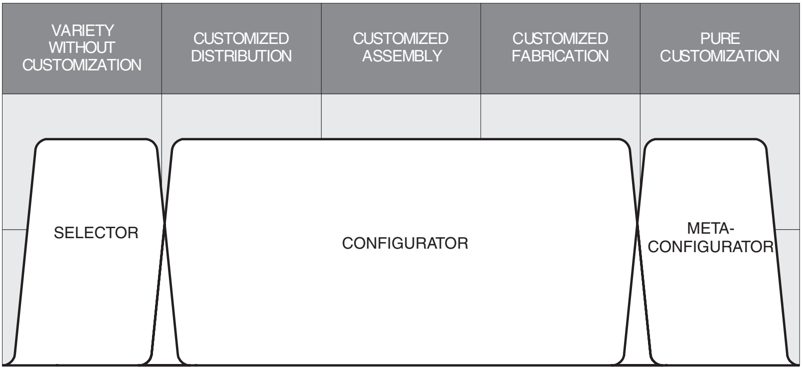


Figure 4.10: Application scope of selectors, configurators and meta-configurators in relation to the scope of customization of the company

resolution algorithms. Such algorithms select the rules to be overridden based on a hierarchy of constraints, and provide a tentative solution to the designer.

To sum up, it is possible to distinguish between configurators and other applications which are somehow similar, such as product selectors and meta-configurators, in reference to the categories introduced in Chapter 1: variety without customization, customized distribution, customized assembly, customized fabrication, pure customization.

As shown in Figure 4.10, while product configurators necessarily occupy the three central categories – categories that deal with the problem of product configuration – product selectors are appropriate in the situation of variety without customization, and meta-configurators in the case of pure customization. Moreover, meta-configurators can be used in any kind of situation, as long as they intervene in the creation of a new product, acting as instruments that support the product design.

5 Commercial Product Modelling

It is essential for a configuration system to support the commercial configuration process. An adequate sales-oriented description of the product family is the cornerstone of any automated support to commercial configuration. Numerous guidelines help in designing such sales-oriented description, the ultimate goal being the reduction of the cognitive complexity associated with commercial configuration tasks. The present chapter:

- conceptualizes the main factors that affect the cognitive complexity of commercial configurations tasks;
- illustrates a number of guidelines to present a company's product offer and to collect the customer's choices;
- explains the mechanisms used to reduce the complexity of commercial configuration tasks and trigger the customer's buying decision.

5.1 Cognitive complexity and configurable products

To offer many product variants may cause a number of difficulties of a design, production and logistic nature, as well as commercial problems. In order to understand this fact, it is convenient to remember that the decision to differentiate or customize the product is based on the assumption that, if the company offers products that satisfy the specific needs of each customer, it is possible to raise sales prices and/or increase the market share of the company. However, in order to achieve these goals, the customer must effectively identify, among the different variants offered by the company, the option that satisfies his requirements. Let us consider, for example, the difficulties we have when selecting clothing in a big department store: the proliferation of different models, colours, trademarks, etc. can exert a negative influence and become an obstacle to our decision to purchase.

This kind of problem is not exclusive to the clothing sector. On the contrary, it may appear every time we have to select the product variant that suits our preferences among 'n' possible variants. In this case, the buyer faces what is called *cognitive complexity*; he faces difficulties in understanding what is offered and, in particular, how the products differ in terms of utility. Even in the presence of only three alternatives (A, B and C) it may difficult to decide, considering their utility, which one is the best. 'What characteristics of A satisfy my needs better than those of B or C? If in some aspects A seems better than B, but in some others the contrary is true, which one should I choose?' This kind of cognitive obstacle is a problem for the company that adopts a strategy of variety, since such variety may confuse the customer rather than convince him that the company can satisfy his need.

Even if we admit that our potential client does not have great difficulties in determining whether A is better than B or C, the problem of cognitive complexity is still not completely solved. Once the customer has decided that, for example, variant B is better than A and C, there is still a doubt: 'Isn't there, among all the other variants offered, one that could be better than B?' When the variants are extremely numerous, to compare them as we did with A, B and C becomes an almost impossible and costly task. The risk is that the customer may delay his decision to purchase, not because he finds it complicated to choose among the variants, but because he does not have a complete knowledge of all the possible alternatives.

In summary, this is a paradoxical situation. On the one hand, the company increases the product variety in order to capture a greater number of customers or to obtain higher prices. On the other hand, the decision to offer a great variety may mislead the client or make the process of selection so difficult that finally the client gives up. The problem of cognitive complexity is well known by the companies that offer variety and customization. Generally, the solution is based on the sales force, which can make the acquisition process easier for the client. Coming back to the example of the clothing department store, the role of the shop assistant is to help the customer find, among the chaos of garments offered, the variant that best suits his requirements. But, why is the shop assistant useful in identifying what we are looking for? To answer this apparently simple question, we must compare our behaviour in two different situations: (1) when we are alone, looking for the product, and (2) when the shop assistant helps us to look for the product.

In the first case, we examine, sequentially and systematically, the garments in the way they are displayed (obviously in the corresponding area). At a glance, we quickly verify whether each piece of clothing may suit our needs.

In the second case, instead, we tell the shop assistant what we are looking for, adding a series of attributes: a shirt with vivid colors, made of cotton, button-down, backstitched, etc. The shop assistant considers the options the shop can offer, and then proposes to us a number of shirts. In other words, he minimizes the complexity we are supposed to face when we want to understand what the shop is offering. We already know that the type of shirt that interests us – if the assistant has not forgotten any option – is in front of us, thanks to the previous selection made by the salesman. The complexity of the selection process was eliminated because the variety of product was presented by *attributes* (colour, stitch, fabrics, etc.) rather than by *alternatives* (shirt #1, #2, … #87 from a shelf ten metres long) and because we trusted the shop assistant for a first selection of the products offered by the department store.

Case 5.1 Selection of a sofa by alternatives or by attributes: an experiment

To understand how the representation of a set of product variants influences the cognitive complexity in the selection process of a potential customer, two

Table 5.1: Representation of the offer in terms of attributes or in terms of alternatives and cognitive complexity (the points not written in percentages range from 1 to 9)

	Variety in terms of attributes	*Variety in terms of alternatives*
1. Are you satisfied by the way in which you could describe the offer of sofas?	4,7	4,0
2. Do you think that the set of variants seems complex?	3,2	3,7
3. Do you think you have enough information to make a choice?	4,3	3,6
4. Do you feel ready to make a choice?	65%	39%
5. Do you feel satisfied with the choice made?	5,2	5,0

marketing researchers selected 78 students for an experiment. They had to choose a sofa for the house where they would live after graduating. In the first part of the test, the students had to understand the product offer. The first group had to consider the eighteen attributes that differentiated the sofas and the 'levels' of these attributes, i.e. their possible values. Supposing, for example, that the colour of the fabric was an attribute that differentiated the sofas, then the relative levels could be: red, blue and brown. The second group, on the other hand, had to take into account a series of possible alternatives. Then the students had to answer a series of questions. Their answers would evaluate their understanding of the offer (see items 1–4 in Table 5.1). Finally, the participants had to choose one of the available variants (see item 5 in Table 5.1).

As shown in Table 5.1, the representation of the offer in terms of attributes, rather than in terms of alternatives, helps the customer's understanding of the offer and enables him to express his specific requirements and choose a satisfactory variant.

This experiment demonstrates that the description of product varieties in terms of attributes, rather than in terms of alternatives: (1) allows the company to transmit, with considerable efficiency, the offer to its customers and (2) enables the customer to express, in a more precise way, his or her own requirements.

As mentioned before, the sales personnel typically carry out this activity of 'educating the customer' or of 'support in selection'. The learning process, however, may be long and repetitive, involving high commercial costs. There is still the problem of how to enable the customer to 'teach himself' or 'auto-configure the product', at least partially. To fulfil this aim, it is necessary to design a supporting program that describes the product, from the commercial point of view. The planning of this support requires a series of fundamental actions and choices:

- choose methods for product description;
- delimit the space of the possible choices available to the customer;
- communicate how different options may create value for the customer;
- structure the ways in which the customer learns and/or defines product characteristics;
- predetermine how the interaction between customer and commercial support facilitates the learning process and, therefore, minimizes the cognitive complexity faced by the customer.

All these points will be analysed in the following paragraphs.

5.2 Describing the product

There is not only one way of representing a generic product to the customer. The product complexity, its importance to the customer and the customer's willingness to learn about the product, are all factors that contribute to determining the most suitable method to describe the product for commercial configuration purposes. To understand how different descriptions of the same product differ from one another, we may consider the simple case of automobiles (see Figure 5.1).

Some of the possible buyers of an automobile are interested in certain fundamental performance factors, but do not care about which functions or systems it has. For example, a grandmother who wants to change her car or the head of a family who is not interested in reading specialized magazines, may only ask for an 'economical and safe car'. A more expert buyer may require an automobile with ABS or traction control. In other words, the customer speaks in terms of functionality, rather than in terms of performance: the wheels should not be blocked when braking or accelerating. A more expert buyer still, one who knows how the car is made, may ask for an automobile assembled with a special kind of tyres, to enhance grip, or he may prefer a model with auto-cooling disk brakes rather than with the traditional single disk brakes. In the case of a really expert driver, he may even ask the concessionaire to install a switch to deactivate the ABS system when very

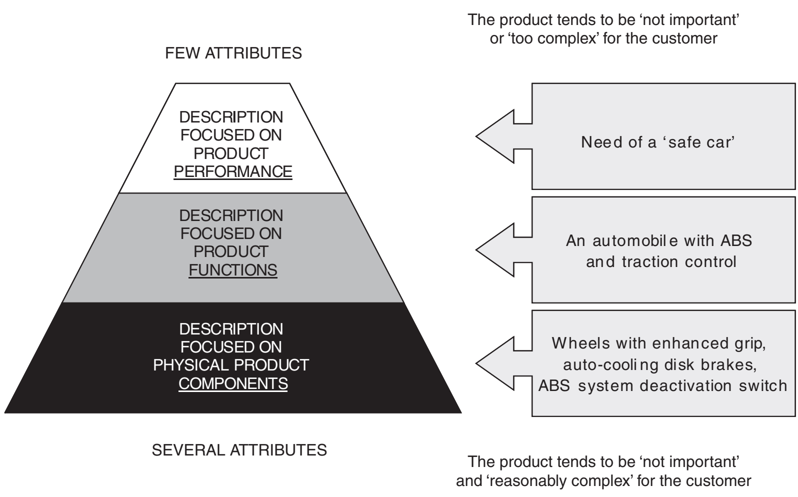


Figure 5.1: Product descriptions with different degrees of abstraction

skillful driving is needed, for example under bad road conditions, due to snow or frost. The third type of customer, in sum, describes the product in terms of components rather than in terms of functions.

The example of the automobile shows us how the same product can be described in very different ways. Different product descriptions can be lined up on a one-dimension continuum. On one end of the line, we can place synthetic descriptions focused on performance, and on the other end detailed descriptions focused on product components. In the intermediate position we find the descriptions focused on functions.

It is important to notice that the three situations are compatible with product descriptions based on attributes. The change is in the nature of the attribute: performance, function or components.

Case 5.2 An example of different product descriptions: the case of computers

An example of different product descriptions is given by the notebook selector of ZDnet (see www.ZDnet.com) and by the Chl (see www.chl.com) personal computer configurator. In the first case, it is not a configurator, but a selector

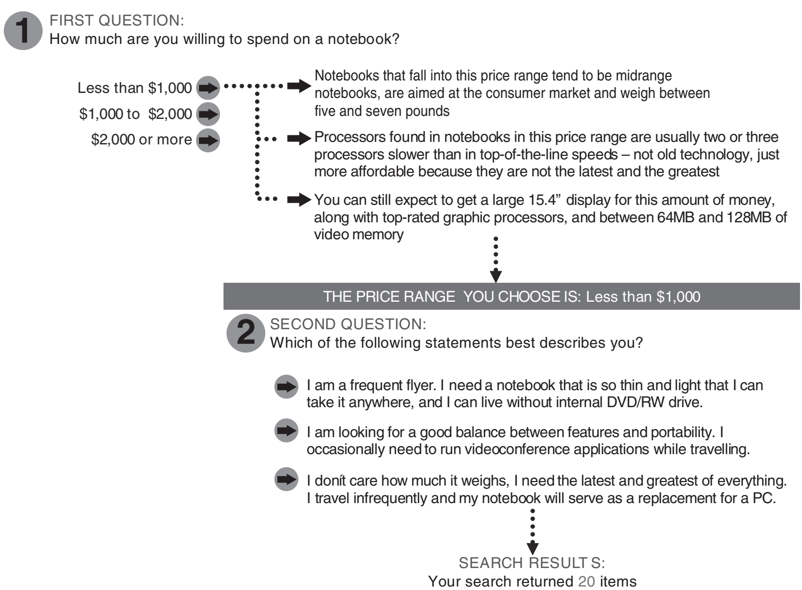


Figure 5.2: Commercial description of a PC with a high level of abstraction

that, starting from a series of characteristics, singles out one or more suitable product variants. Anyway, as stated in Chapter 4, the elaboration of a commercial model or of a formal description of the product, from the commercial point of view, are activities performed by the selector or by the product configurator. In the case of ZDnet (www.ZDnet.com) the company has conceived a product description at a very abstract level, to help the inexpert potential customer to choose among the numerous variants offered. The selector asks general questions such as: How much are you willing to spend? How often do you take a plane, a situation where weight is an important factor? If weight is not a problem, are you looking for a notebook that can be used as a desktop replacement? In Figure 5.2 we see how these simple questions, that anyone can answer, allow the customer to skim the offer, reducing the number of options to a dozen potential products.

In contrast, we find the personal computer configurator created by Chl (www.chl.com). In this case the product description is made at the level of single physical components, that must be entirely specified, from the computer case and the motherboard to the possible peripheral units. As shown in Figure 5.3, it is necessary to have considerable knowledge of the components and functions of a personal computer, to get an effective advantage from the possibility of expressing one's preferences at a product component level.

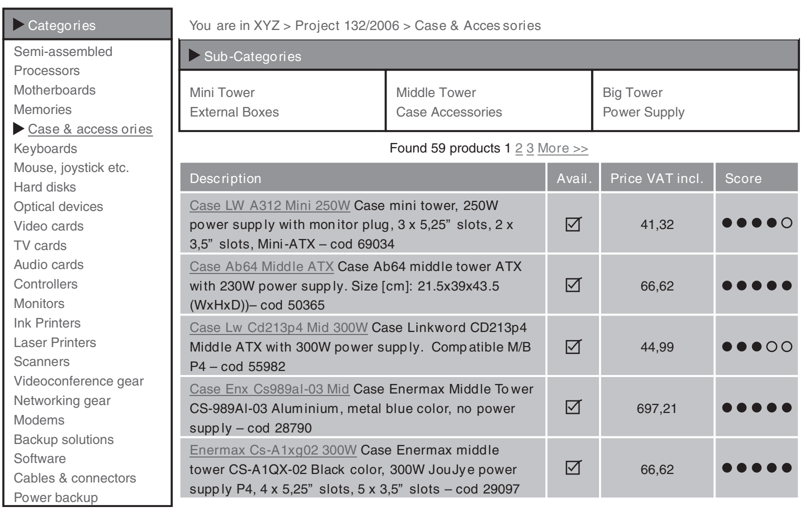


Figure 5.3: Commercial description of a product with a low level of abstraction

Choosing the appropriate way to describe a product is not a question of using a language that could be more or less abstract. In some cases, the characteristics of the product require an input for the commercial configuration that has a format different from the textual one considered up to now. For example, a company that sells cooling systems for cold rooms has as a fundamental input for the commercial configuration process: the layout of the cold room; the plan of the room; free walls on which the cooling fins can be placed; and possible obstacles for the installation. A description in terms of measures, using a standard format to collect the necessary information, would be practically impossible! In general, the problem of getting the characteristics in the form of a layout is common to all the companies that manufacture industrial facilities and systems. The alternatives for these companies are two: (1) to limit the configuration only to facilities or systems components, leaving the layout definition to sales engineering; (2) to invest in information tools that allow for layout specification too. In this case, the company has to compare the benefits of automation very carefully with the costs for the development of a software solution that, in all probability, will require a certain degree of customization, and therefore a further investment.

Another important aspect of commercial configuration is the total description of the configured variant. Obviously, this requirement is based on the fact that the final aim of a commercial configuration is to carry out a business transaction. Price and an indicative delivery time are fundamental outputs of the commercial configuration process. Defining all the aspects of the product is very important to avoid misunderstandings, opportunistic behaviours, disputes, etc. Furthermore, it is essential to give a complete vision of the product ordered by the potential customer, because in this way he is able to verify at a glance the whole effect of his preferences, considering even some interactions that may be neglected due to his choice by attributes. Let us consider, for example, the case of the sofa, described in section 5.1. If we offer the client the possibility of choosing the type of wood needed for some parts of the sofa and the colour of the fabric, surely he will be interested in knowing whether both colours, put together, produce a pleasant match or not. This problem could have been avoided if the choice had been made by alternatives, showing each one with its corresponding illustration. This kind of service is offered by the websites of some automakers, where customers can simulate the matching between the colours of the interior and the body, avoiding unpleasant impressions when the automobile is taken from the car showroom.

5.3 Limiting options

Even if the most suitable way to describe the product has been determined, the problems associated with the definition of the commercial model have not finished yet. A typical dilemma is to decide whether to include in the commercial model all the possible variants. From a production point of view, it is important to remember that, even if the various components needed to create an offer of 'n' products have already been designed, to include all of them in the commercial model means assuring the supply to a possible customer of a variant that is very rarely manufactured. This may bring about a series of difficulties in the supplying, planning and control of the production activities, with the risk that the costs may be higher than the revenues obtained by selling that kind of variant. In sum, it is necessary to remember that behind a commercial model there is always a workshop that has to manufacture the product!

Furthermore, the reasons for limiting the variants among which the customer can choose, in relation to the ones the company can offer, may be purely commercial in nature. In fact, *to offer many variants, in the end, always complicates the commercial model* as well as the customer's choices. Sometimes it can be more practical to label some 'exotic' options as not available, in order to reduce the quantity of information the customer has to supply to obtain a complete configuration (see Figure 5.4). The simplification of the configuration dialogue – a further confirmation of the problems associated with the cognitive complexity of the configuration tasks – is one of the requirements most frequently underlined by the commercial department of the companies that are implementing a product configurator. The customer who asked for something that is not foreseen by the product model should get in touch with a salesman and check the possibility of obtaining the requested variant.

In other cases, the decision to delimit the variants among which the customer can choose is a consequence of the rationalization of the

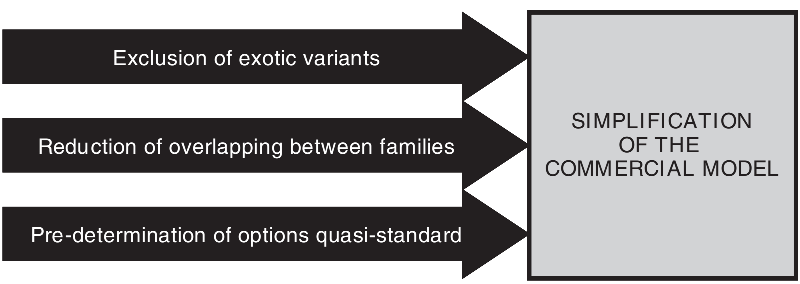


Figure 5.4: Simplification of the commercial model by limiting the options

offer the company makes, when it decides to structure the configuration process. Very frequently, the company 'realizes' that different product families, in some way, tend to overlap (Figure 5.4). The most expensive variants of the family at 'entry level' or with 'low power' overlap with the cheapest variants of the immediately superior family. This may happen even in the most famous companies. Boeing, for example, expanding the capacity of the 737 family and developing more spacious and comfortable versions of the 767 family, finished up by covering, with these two families, the field of the 757 (that never actually 'took off'!) In general, if the overlapping of product families is not commercially justified, the formalization of the dialogue should reasonably lead towards a limitation of the variety offered, in relation to the variety the company is potentially able to offer (and in the future towards a new definition of the latter).

A third possibility to reduce the quantity of commercial characteristics to be specified in the product configuration is the predetermination of some characteristics that generally are requested in a certain standard (Figure 5.4): in other words, to foresee default values for some product characteristics that rarely acquire values different from those pre-established. An extreme case of this approach is to offer the customer a set of pre-configured products, with the possibility of modifying the attributes according to his preferences, as long as he respects the constraints defined in the commercial model.

Case 5.3 Limiting the options available in the commercial configuration of a gas regulator

Silentwhistle, a well-known company that operates in the sector of gas reduction and control, carried out research on the possibility of automating the configuration process of its regulators. The analysis of the commercial configuration process showed that the process of selection on the part of the customer could be simplified by using the three limitations mentioned in the previous paragraphs. For example, the company managed the 'exotic' variants by means of a 'note' field in the paper form used for the configuration. The problem arose because the same 'note' field was also used to specify necessary attributes such as the regulator adjustment. This approach is not suitable for an automatic management of the product variants. The solution to this problem is the elimination of the 'note' and the creation of items for the adjustment or other 'frequent' specifications. As far as 'exotic' requests are concerned, the customer finds, in the configuration dialogue, the commercial office free phone number, to be contacted in case he wants to check the feasibility of what is not foreseen in the sales dialogue.

Moreover, in the creation of the commercial model, there appear to be many difficulties associated with the design of a family selector, that can guide the customer towards the configuration of a regulator within a specific family. The families elaborated by the company, in fact, present many overlaps and there is no clear criterion for the identification of the family that best satisfies the potential customer's requirements.

Finally, some items, for example the type of gas, are standard in 99 per cent of all cases since nearly all the installations use natural gas. Obviously, it is pointless to ask the customer to specify a value for such an attribute every time he orders a product. In any case, the customer still has the opportunity to specify a possible difference from the standard.

5.4 Communicating value

As mentioned before, any kind of strategy of product proliferation develops different options that create different functions, as a more precise answer to the preferences expressed by different customers. To communicate the value of such options, we can certainly describe the product attributes using a language that the customer can easily understand, and we can avoid overwhelming him with too many options, but this is not enough. Maybe the customer simply wants to understand the product or the way in which the different attributes determine the functionality of the product. This is a typical activity performed by the commercial staff. It is a complex function, because it implies a certain 'didactic' skill on the part of the salesman, and because the learning process on the part of the customer may be long. Moreover, and above all in the case of products that are complex and/or subjected to rapid technological evolution, it is necessary to teach the different product functions not only to the customer, but also to the salesman. To develop automated solutions to illustrate the functions associated with the different product attributes, on the one hand, and to enable the client, as well as the salesman, to learn autonomously, on the other hand, the learning process is carried out according to the modes and times of the user – perhaps when he is relaxed at home or during a calm moment at work. Better knowledge will give the customer better elements to decide about the product he is willing to buy.

If the product presented by the company is neither valid nor suitable for the customer, the company that implements this approach will lose a potential client. But, if the product is not suitable and the customer decides to buy it, in the end the company will also lose him.

The progress of multimedia and the growing bandwidth of telecommunication services offer a number of promising opportunities to the companies that want to communicate the value of their product varieties. Films, animations, graphics and sounds help to reduce the time the customer needs to understand the complexity of a product family

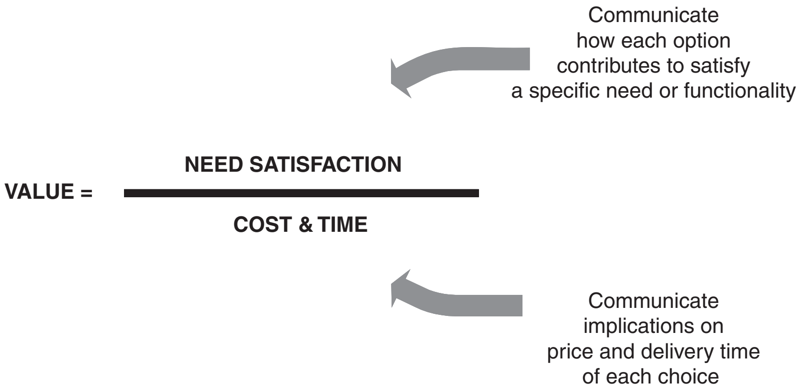


Figure 5.5: Fundamental activities to communicate the value of the variety offered by the company

and consequently increase the profitability of product proliferation strategies.

Communicating the value of the variety offered to the customer does not only mean explaining what the different alternatives are able to do (Figure 5.5). Not all alternatives are so easy to elaborate. Some may require the design of ad hoc components, and are part of a field that is outside the product configuration process. But, for commercial reasons, companies cannot exclude from their product offer those semi-configured or particularly problematic variants. In these cases, to communicate the value of his/her specific choice to the client, also means *making him aware that he is asking for something 'special',* and that his requirement will probably influence the price and/or delivery times.

Case 5.4 Communicating the value of the product options: the case of Dell Computers

Dell Computers, one of the first companies in the world to sell configurable products on-line and one of the biggest producers of PCs and portables, provides an example of 'customer education' aimed at optimizing the utility of the available options that is perceived by the client. As shown in Figure 5.6, if we take, for example, the option of hard drives in the main configuration dialogue, we see that there are four variants.

If the customer is not yet ready to decide, he can ask for more information (learn more). At this point he is able to consult three windows. Two of them have a purely educative function: one explains the primary characteristics of hard drives and how they determine functionality, the other describes one by one all the technical attributes of the hard drive. The third window combines the generic description of hard drive attributes with the specific hard drives offered by the company.

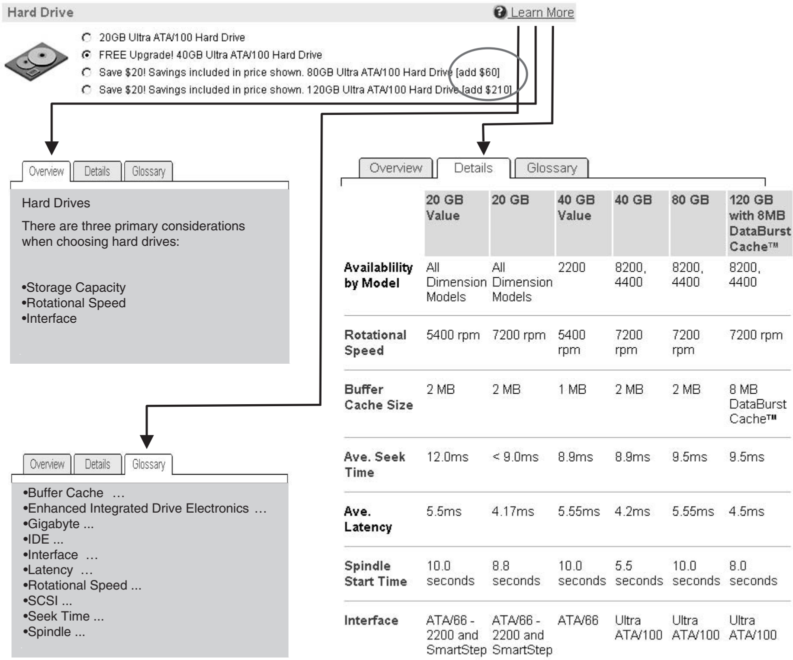


Figure 5.6: Communication of functions related to three hard drive variants (adapted from dell.com)

At this point, the customer is able of evaluating the different profiles of attributes of available hard disks, picking the one that best approximates his 'ideal' attribute profile. Finally, but not of minor importance, another benefit of customer education is that the potential client, after getting an idea of what product could satisfy his needs, can go back to the main menu and evaluate if price variations are justified in relation to his needs.

5.5 Structuring the process of interaction with the customer

The elaboration of a commercial model and the definition of how the sales dialogue will be carried out raises – apart from the problems mentioned in the previous paragraphs – the issue of how to structure such interaction. In the simplest case, the answer to this question implies the definition of the order according to which the different questions are asked. A company that wants to communicate to its customers the idea that it offers strong customization, for example, may consider it useful to present, in the first place, the product attributes that offer more possibilities of choice. For a company that sells tailored shirts, the type of pattern (checked, striped, plain, etc.) and the different variants (big or small squares, tartan, etc.) will surely be the first questions to ask. In other cases, especially for technical products, the sequence of questions has to follow, as close as possible, the order the customer typically uses when describing or specifying the product. For example, a company that offers customized pumps, firstly, will ask for the type of application (submergible, peripheral, etc.) and then for some fundamental data such as flow rate, discharge head and so on.

In general, the idea is to *allow the customer to search the variety offered by the company in the most natural and spontaneous possible way*. The importance of this condition is not reduced if the user is the salesman. In fact, if the salesman perceives the process followed to define the product characteristics as something complicated and unnatural, the chances of implementing a successful solution, even if partially automated, are very low.

The elaboration of a structured process that leads the customer, probably with the help of the salesman, towards the definition of a commercial configuration, presents some difficulties that derive, not from the customer, but from constraints between different product attributes. For example, in the case of a sub-compact car, the option 'air conditioning' may not be available for the version with a reduced displacement, due to the fact that excessive power will be absorbed by the compressor from the crankshaft.

The presence of links between options determines a set of constraints in the sales dialogue. For example, *some questions must be asked following a fixed sequence*. Let us take the case of a scooter: if we want to configure a scooter with the options 'country' and 'double-seat', we must choose first the option 'country' and then the option 'double-seat', because in some countries two people are not allowed to ride on the same motorcycle while in others this is permitted. A second rigid condition, related to the first one, is the fact that *possible choices of a certain attribute depend on previous choices of other attributes*. For example, if in the configuration of a bicycle, we have selected the options 'titanium alloy frame' and 'double damper', it is evident that the choice of different types of rims will be limited to those that respect or surpass a certain minimum value of strength. In some cases, these limitations finally eliminate any kind of freedom while choosing certain attributes: only one of the levels admitted for a certain attribute could be compatible with the choices previously made in the sales dialogue.

However, the presence of links between product attributes on which the customer can express his preferences, generally does not exclude the possibility of defining various alternatives. Let us consider again

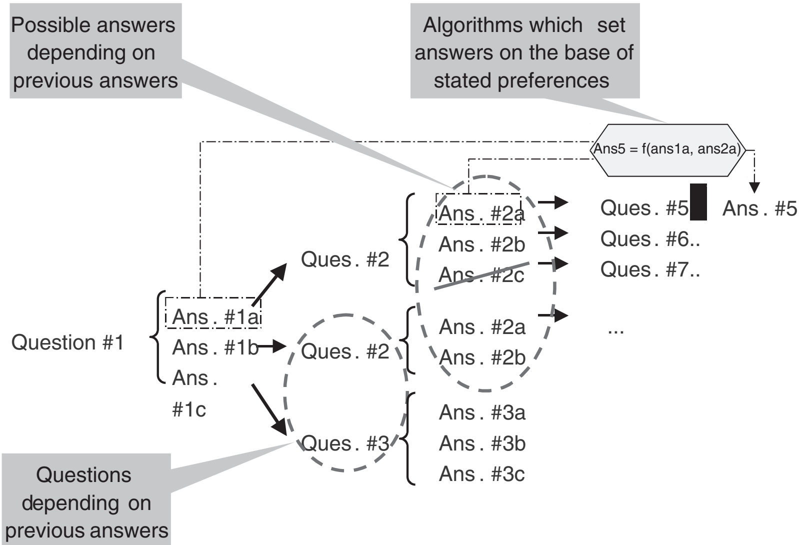


Figure 5.7: Structure of the sales dialogue

the example of the bicycle. For the attributes 'type of frame' and 'weight' the customer could specify the maximum weight, and so a series of 'heavy' component variants would be automatically excluded from the choice (frames of chromo-molybdenum steel, standard saddles and other elements). On the other hand, the weight could be simply calculated by adding all the weights of the components selected. In this case, in order to reach the required weight, it may be necessary to modify the choices made. It would be interesting to allow the customer to start the sales dialogue from any compatible attribute of the product. Yet, this complex option is rarely found in commercial applications. In fact, the most common approach consists in considering a precise sequence of questions and answers (Figure 5.7), in which we can have:

- questions that depend on the previous answers;
- possible answers that depend on the previous answers;
- algorithms that develop answers on the basis of previous preferences.

To represent the sales dialogue as a decision tree is a conceptual approach to the design of the commercial model. The way in which this representation will be implemented in the product configurator obviously depends on the particular software system adopted.

Methods for product description, delimitation of a customer's choices, communication of the value related to the different options and, finally, structure of the sales dialogue, sum up the actions needed to build a good commercial product. We still have to consider the fact that a company has to serve, with the same product family, many different types of customers (let us take the example of automobile factories). The question is: Is it possible and efficient to serve all these types of clients using a single commercial model and, consequently, using a single sales dialogue? Or is it better to develop different commercial models according to different types of customers? There is no one correct answer. First, it is fundamental to consider marketing aspects and the investment necessary to develop different commercial models, and then to determine whether and to what extent one solution is more appropriate than the other.

5.6 Interaction and learning

All the guidelines discussed up to now basically aim at reducing the complexity perceived by the customer, alleviating the information processing load he undergoes when assessing the company's product offer. However, there is another way to achieve the same goal, i.e. to increase the customer's ability to understand and evaluate the available product information. Learning, therefore, is a fundamental activity in reducing cognitive complexity.

Learning, however, is a process, i.e. it consists of a series of actions carried out over a period of time. Traditionally, such a customer learning process has constantly been supported by salesmen. The salesman feeds the customer's learning process with repeated visits, showing him product sales documentation, and communicating with him in multiple ways.

A precondition for a successful automated commercial configuration tool, accordingly, is the capacity to support the customer's learning process. Fortunately, interactivity is, in some ways, an intrinsic characteristic of computer applications. Because of this interactivity it is possible to use the sales dialogue as a basis for educating and updating the customer, as far as the company product offer is concerned. In order to understand the product, the sales dialogue can be gone over several times, giving different answers to see how the final product configuration changes. A better understanding of the product eventually enables the customer to appreciate how different possible attributes contribute to create a configured product variant that can satisfy his requirements.

Case 5.5 Interactive learning of the interdependencies between different characteristics of a truck

The configuration of a truck offers an interesting opportunity to understand the fundamental role played by interaction in the comprehension of how the choices we make are interdependent (product knowledge) and how such choices influence the general performance of the product. The truck studied presents a considerable number of possibilities: cabin type (7), engine (4), class (4), chassis type (2), wheel configuration (11), chassis height (4), suspension (3), power (11). Furthermore, the client can express his preferences in terms of maximum price, level of comfort, versatility, performance and fuel economy (see Figure 5.8). All these items are interdependent.

Let us start from the configuration shown in Figure 5.8, characterized by a 'normal' level of comfort. Let us suppose that we want to purchase a more comfortable vehicle, so we specify a 'very high' level of comfort. Immediately the system offers us a modified outlook of the performance, different from the one shown in Figure 5.8 (see Figure 5.9). In the new table we can see that the increase in the level of comfort implies a decrease in the level of performance (Low) while the level of versatility does not change (Normal) and the level of fuel economy is the same (High). The price is 15 per cent higher, from 566 070 it is now 649 620 (Swedish crown).

At this point, it is interesting to see what happened at the level of components, that is to say which components guarantee high performance in terms of comfort. As seen in Figure 5.10, the option of very high comfort has modified the cabin type (CP19 instead of CT14), the chassis height (low instead of normal) and the suspension (air instead of leaf suspension).

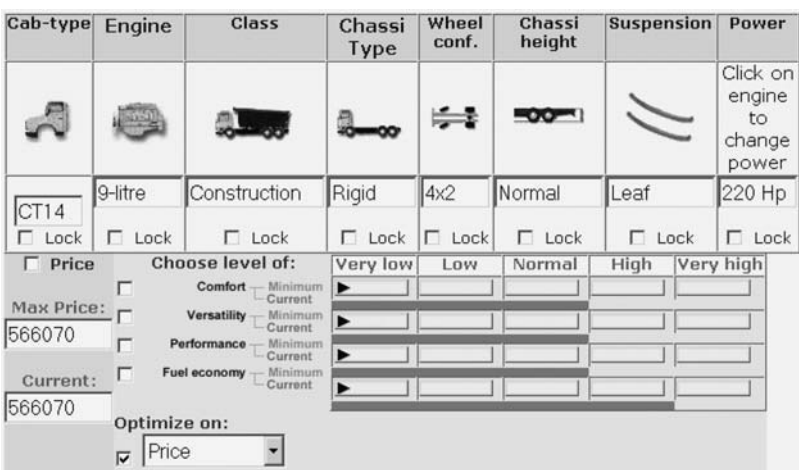


Figure 5.8: Attributes for the configuration of a truck (adapted from Tacton.com)

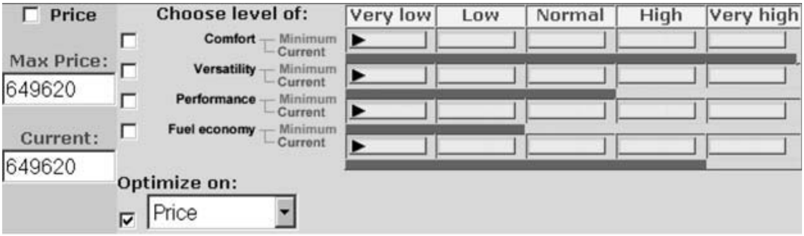


Figure 5.9: Interdependence between parameters (adapted from Tacton.com)

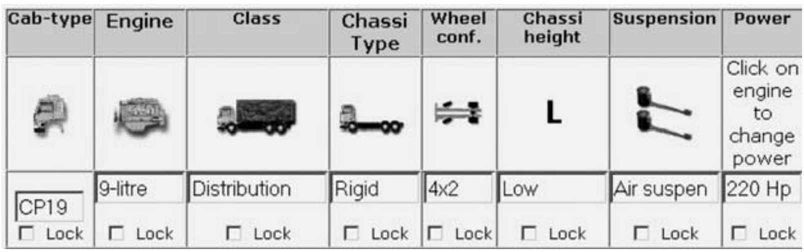


Figure 5.10: Modifications of some components due to the specification ‘maximum comfort level’ (adapted from Tacton.com)

In order to configure the truck, the system took into account the fact that the price was specified as a criterion of optimization (bottom left box ‘optimize price’). The system chose the smallest engine compatible with the specifications supplied (220 Hp), reducing the general performance. If we click on ‘engine’ we see that there are two other alternatives, compatible with the configuration chosen, engines 260 Hp and 310 Hp. If we select the latter, we get a truck that costs 655 170, with an increase of 0.8 per cent. The system also informs us that it is possible to choose other engines, but this option may modify other characteristics of the truck. For example, if we select a more powerful engine, 400 Hp, the cabin type becomes CR19, the class ‘long-distance’ and the chassis height ‘normal’. Exploring these interdependencies, the potential customer can get a clear idea of what the company offers, and above all, of what the company can do for him. We must remember that a family of heavy trucks may have thousands of possible configurations, considering the detailed specifications.

5.7 Putting everything together

At this point, it is possible to summarize the activities that lead towards a structured knowledge of the product, from the commercial point of view, and towards the building of a good commercial model and finally, of an efficient sales dialogue. The starting point is, as

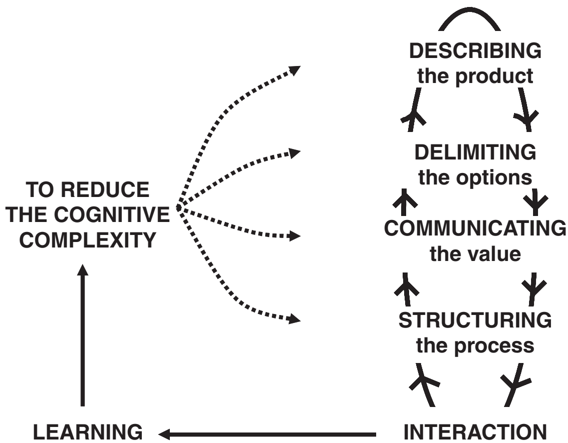


Figure 5.11: Integrated scheme for a conceptual design of the commercial model

repeatedly mentioned before, to reduce the cognitive complexity related to the efforts the potential customer has to make, in order to understand what the company offers and how the different solutions can satisfy his requirements. The main lines to project the interaction with the customer are: 1. describing the product, 2. delimiting the options the customer can make, 3. communicating the value of the different alternatives, 4. structuring the customer's process of interaction (Figure 5.11). The opportunity for the customer to interact directly with the commercial model or through the salesman, generates a learning process that facilitates the understanding of the offer and relates it to the customer's specific needs.

6
Technical Product Modelling

The description, from the technical point of view, of each product variant is a distinctive function of configuration systems. The performance of this key function requires, on the part of the company, the elaboration of a large amount of technical information. Consequently, the knowledge required to process such technical information has to be captured within appropriate logical structures. The present chapter:

- summarizes the main difficulties associated with the organization of technical information, in the case of configurable products;
- explains the 'traditional' approaches used to describe product variants from the technical point of view, illustrating their limitations;
- discusses the concept of a generic bill of materials, i.e. the logical structure that generally implements the technical model;
- supplies guidelines for designing generic bills of materials.

6.1 Technical description of configurable products

As seen in the previous chapters, a fundamental aspect of product configuration consists in generating the necessary information to describe how each product variant is physically obtained and how it is actually composed. It is clear that the description of a product is a fundamental activity for the industrial production of goods. A craftsman might have in his mind all the information necessary to manufacture a product, something impossible in the case of an industrial company. Without a formal description of the product, the different departments in charge of designing the product and the production process, of supplying the necessary material and of programming production, would not be able to work in a coordinated and efficient way.

One of the most common instruments used to describe the structure of a product is the *bill of materials* (BOM). The bill of materials describes the product as a tree, in which the offspring of each parent are all and solely that one parent's components. Therefore, it is described by two types of data: the BOM *elements*, characterized by attributes such as code, description, etc., and BOM *relationships* characterized by parent code, son code, quantity, etc. Very often, it is necessary to integrate this information, which describes the product, with some information which describes how the product is made, that is to say the production cycle. A third fundamental type of datum is the *cycle element*. A cycle element specifies which operations are needed to make a given parent from its component(s), as indicated by the bill of materials. In describing such operations, information on what resources are needed as well as their cost is typically included (see Figure 6.1).

Immediately, we notice how the difficulties, generally associated with the collection of product documentation, increase if the company offers many product variants or customized products. In fact, a proliferation of product variants necessarily leads to a proliferation of the information needed to provide the documentation of the products manufactured by the company. It is not unusual for a company offering customized products to have to handle hundreds and even thousands of codes and even more BOM linkages. The possibilities of mistakes and the presence of a great quantity of 'dead' codes are considerably high. Under such conditions, keeping correct information

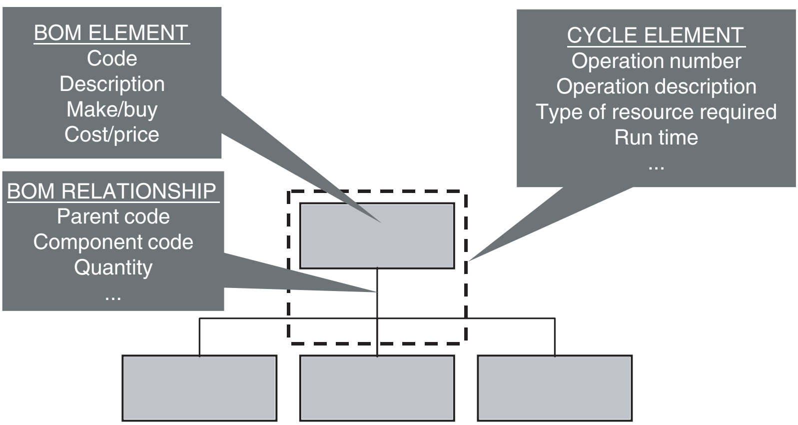


Figure 6.1: Information needed to describe each product variant

about production cycles is particularly difficult and production scheduling becomes unreliable.

The problems associated with the handling of product information, in the presence of a considerable number of product variants, are not only due to the enormous quantity of information needed to describe all the product variants. For example, while describing a truck, the designers consider the gear system as a single 'gear group', formed by the gear box, the gear controls in the cabin, etc. But from the production point of view, the gear system is divided, at least, into components belonging to the 'engine group' and components belonging to the 'cabin group', since these groups are assembled in different sections of the factory. On the one hand, to see the components of the 'gear group' as a whole, helps the technical office to check the functioning of the gear system. On the other hand, to see the components of the 'gear group' as a whole along with other physical elements, helps oper-

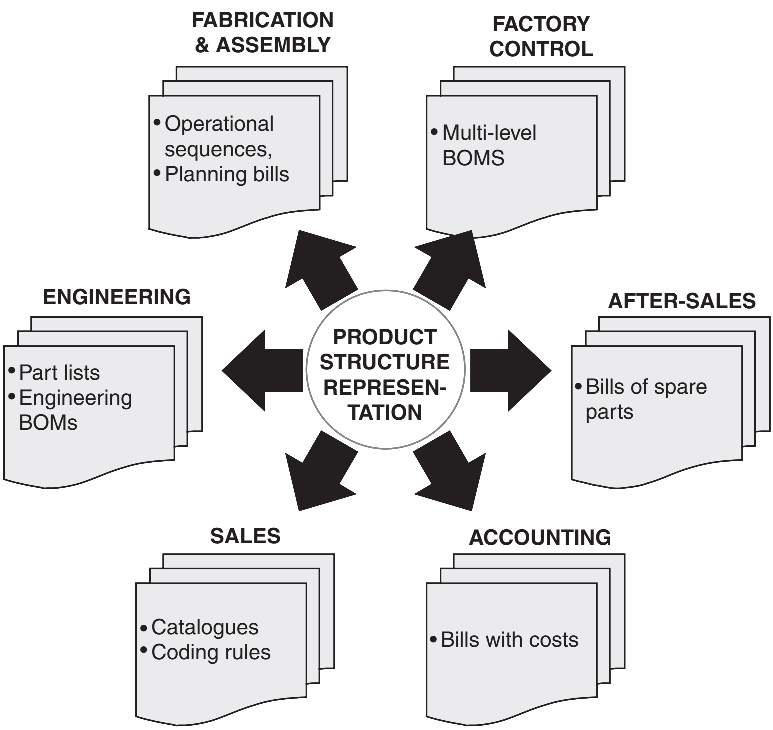


Figure 6.2: Different views of product structure

ations to manage sourcing and assembly activities. It is obvious that when there are ten variants of gear systems, instead of one, the tasks necessary to create and update these different 'views' of the product increase considerably.

As shown in Figure 6.2, different departments need different product representations, even if these representations are, of course, interconnected. Assuring the existence and compatibility of such descriptions, when there is a great number of product variants or a constant possibility of getting new ones, represents a further challenge to the company.

The company that offers configurable products has to face the problem of handling piles of product documentation. On the one hand, the company may take advantage of the fact that the space of the feasible product variant is identified, and prepare the documents for each product variant 'a priori'. In this way, the organization of the product documentation will not delay the order acquisition and fulfilment process. However, in this case, the company incurs considerable costs, in order to maintain the product documentation, especially when countless combinations can generate hundreds or even thousands of possible variants. On the other hand, the company may decide to minimize the costs of handling product documentation, creating bills of material and production cycles only on customer's requests and orders. Consequently, there could be delays in the answers and inefficiency in the elaboration of such documents.

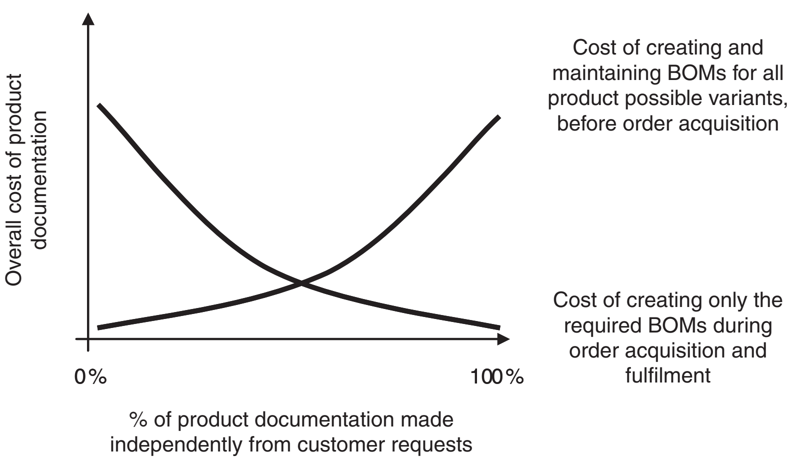


Figure 6.3: Alternative approaches to handling product documentation

Research on product documentation has developed some approaches such as the add-delete bills and modular bills (see section 6.2 and 6.3) that try to conciliate the contrasting aspects illustrated in Figure 6.3. Yet, these documents may be difficult to use and are not suitable in the case of a generic configurable product, which is why they are scarcely employed (see section 6.4). In the presence of configurable products it is possible to adopt a third approach, which occupies an intermediate position between the two extreme situations shown in Figure 6.3. This approach is called the *generic or variant bill of materials* that will be described in section 6.5.

6.2 Add-delete bill of materials

Add-delete bills can be used when the product variants derive from a base product B, for which a code and a bill of materials have been defined. The product variants V1, V2, etc., that technically descend from B, can be defined by simplified bills, called add-delete bills, which show the components that were added to and deleted from B, in order to get each new variant. For example, in Figure 6.4 we notice that variant V1 is obtained by adding the component C and deleting the component E from the base product B.

It is clear that this technique reduces the number of combinations, in relation to the case in which all the product variants are completely described. In fact, if a certain add-delete bill is associated with each commercial option, all the bills related to the different combinations of

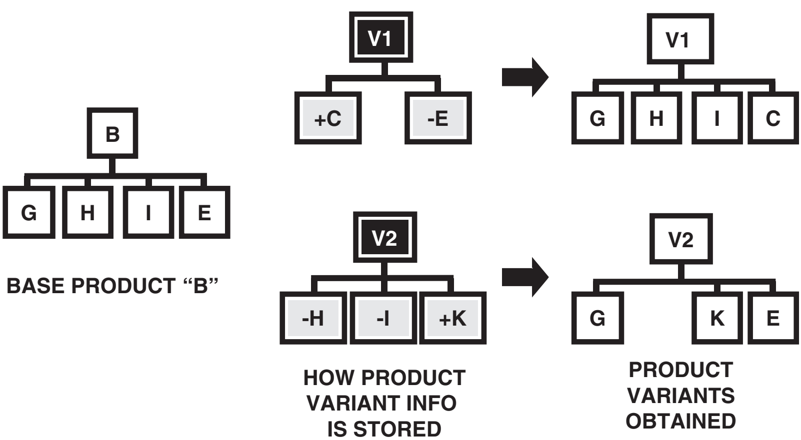


Figure 6.4: Add-delete approach used to describe product variants

options will be automatically generated. Moreover, the add-delete approach can simplify the updating of the bills because, whenever it is necessary to modify a component shared by the different possible variants, it is enough to modify only the bill of the base product. However, the risk of errors is very high, either because the modification of a component of the base product will be applied simultaneously to all the add-delete bills in which such a component is deleted, or because the modification of a component of the base product will require the modification of components added to some or to all the add-delete bills, for example because of the presence of constraints between components. These difficulties, and others, make this approach rather unsatisfactory for product families with a high number of varieties.

Case 6.1 Add-delete approach for the configuration of industrial vehicle variants

A large European manufacturer of industrial vehicles used, some years ago, add-delete bills to describe product variants for commercial, production and logistic purposes. A heavy industrial vehicle can be manufactured in thousands of variants, according to some 'primary' functional characteristics, such as wheelbase or number of axles, etc. or to some 'secondary' optional characteristics. The approach used by the factory consisted in defining a series of base product bills that differ in their 'primary' characteristics, called Vehicle Country Block (VCB). Each commercial option (OPT), for example power steering or heated windows, had an add-delete bill (BZ) that indicated the components (KZk) that were added to or deleted from the base product bill, in order to obtain the variant with the required option. A customer specified a certain base vehicle, for example VCB3, and a series of components, such as OPT32 and OPT12. The bill of the variants required by the customer was obtained by taking the bill of VCB3 {KZ1, KZ2,……KZ134} and adding it to the bills BZ32 {-KZ2, +KZ15 … } and BZ12 {-KZ21, +KZ33 … }.

The first problem of the add-delete approach was related to the fact that it was not possible to have a single base product bill. It was necessary, instead, to refer to a certain number of base models. Due to the high variability of the vehicle, in terms of engine, transmission, cabin, wheelbase, chassis, etc., it was senseless to try to get each variant of a base vehicle, since this base vehicle did not exist. The immediate consequence of so many base models was the redundant information about the product. If there were two types of cabin and three types of engine that could be combined freely, we had 6 VCB. Moreover, if we offered two variants of wheelbase, the VCB increased to 12, with a great quantity of redundant information. This redundancy not only affected the combinations, which could be considered as a 'minor' problem, but also the rules. For example OPT34 was incompatible with engine 1, VCB4 (short cabin) and VCB5 (long cabin); the addition of a new wheelbase and of two new VCB (VCB8 and VCB9) required the specification of two new rules: OPT34 incompatible with VCB8 and with VCB9. In the presence of thousands of vehicle variants, add-delete bills had specified 633 300 configuration rules! At the moment, thanks to a more efficient approach, configurations rules are reduced to 9100.

6.3 Modular bills

Modular bills represent an alternative approach to generating bills of material for product variants. In this case the issue is not to transform a base product into the requested variant, but to create 'sets of components' (modular bills) that will be used to compose a description of the variant. An example will help us to understand the mechanism of modular bills.

Case 6.2 Modular bills for microwave oven: an example

Let us consider a microwave oven that offers two options: colour (white and black) and control panel (analogical and digital). The bills for the four variants are represented in Figure 6.5.

The process for elaborating modular bills requires the identification of:

- components that are common to all variants;
- components that are associated with a certain option (option-sensitive).

In the case of the microwave ovens, we can see that, for example, codes 322, 378 and 300 are common. On the contrary, codes 398 and 365 are associated with the option 'white', codes 376 and 363 with the option 'black', codes 326 and 309 with the option 'analogical control' and codes 334 and 310 with 'digital control'. Therefore, it is possible to build boxes, associated with the different commercial options, called modular bills (see Figure 6.6).

The division of the product into modular bills – that is to say, into groups of common components and option-sensitive ones – presents a series of advan-

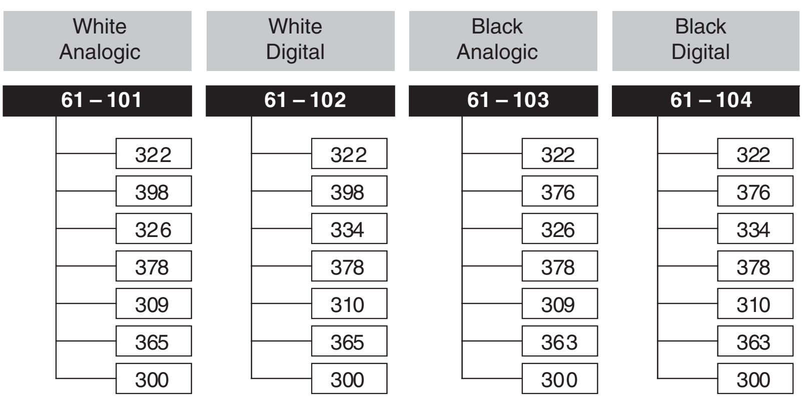


Figure 6.5: Bills of four microwave oven variants (only level 0 and 1)

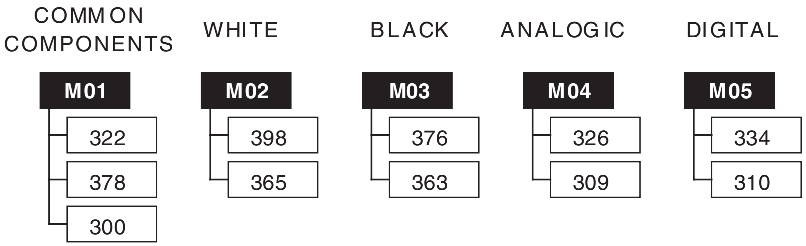


Figure 6.6: Modular bills for a microwave oven

tages. First, pre-vision and planning activities for the production process are simplified, especially when there are many options. In fact, it is easier to foresee the sales by a certain attribute (I think sales will be 30 per cent black ovens and 70 per cent white ovens) rather than by the combination of attributes (I think sales will be twelve black ovens with analogical control, 18 per cent black ovens with digital control, 43 per cent white ovens with analogical control and 27 per cent white ovens with digital control). Then, the updating of the bills is also simplified. If a modification has to be introduced in the analogical control panel, only M04 will be modified. If all the bills were supported by documents, it would be necessary to modify four different bills (in the case of a product such as a microwave oven, very frequently there are at least hundreds of bills!).

The method of modular bills works only if a number of conditions are fulfilled. These conditions have some negative consequences.

First, *the options must be independent*. Coming back to our example of the microwave oven, we can have the option 'thermal heating', which

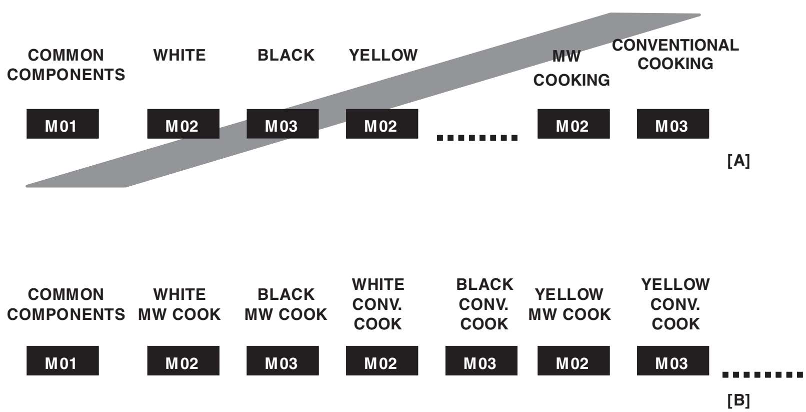


Figure 6.7: Effects of interdependent options on modular bills

implies that the microwave can be used as a conventional electric oven. In this case, the interior of the oven must be enamelled rather than varnished, so that the walls can withstand very high temperatures. If the customer who orders an oven wants, for aesthetic reasons, the interior and the exterior of the oven in the same colour, then the option 'colour' interferes with the option 'type of cooking'. The type of interior to be assembled will depend on both options: 'type of cooking' (enamelled or varnished interior) and colour (white or black). In terms of modular bills the case shown in Figure 6.7 [A] will not be possible and so it will be necessary to adopt the solution of Figure 6.7 [B]. The result of the interdependence of commercial characteristics very frequently causes a proliferation of modular bills, an increase of complexity in the elaboration of such bills and, finally, a progressive abandonment of the use of modular bills. If a product is not modular, the method of modular bills is not particularly helpful.

The second condition is related to the fact that, after getting the customer's acquisition order, *it is necessary to have a complete bill of the variant ordered*. This document is fundamental to planning and controlling the production of the variant. As mentioned in section 6.1, in order to program and control production the bill must be structured in a way similar to that in which the production process will be carried out. This means, for example, that parts of the product elaborated in different departments of the company should be listed in different sections of the modular bill. In the case of the microwave oven, the option 'white-thermal heating', for example, has components that are produced or assembled in different steps of the production cycle. The differentiation of the external part of the oven will be carried out in the 'varnishing' department, while the differentiation of the interior part will take place in the 'enamelling' department and, finally, the differentiation of the type of cooking in the final assembly line. Dividing the product into codes based on option-sensitive components, the information concerning the production process is practically lost – information that becomes fundamental during the order fulfilment process.

6.4 Common practice

Traditional approaches proposed for dealing with documentation related to product variants, as we have seen, present some limitations and complexity that make them difficult to use in many production contexts. The limited diffusion of these approaches, especially in medium and small-sized companies, is mainly due to the rigidity of such approaches. Considering their origins, we can see that traditional approaches originated in the 1960s, in companies such as IBM, with the attempt to develop instruments to support the production process of customers that used mainframe systems. Market micro-segmentation and mass customization had yet to occur. Accordingly, the extent to which product variety and customization were offered was definitely moderate compared to today's way of doing business. The practices and methods conceived to handle such moderate variety, therefore, are definitely unsuitable for those companies that nowadays compete by customizing their products.

Large and medium-sized enterprises reacted to the new competitive challenge by adapting and modifying these old approaches to fit their specific needs and business models. Small-sized enterprises, on the other hand, often relied on what we may call a 'degenerated' add-delete approach. According to this approach, the company produces documents for the requested product variant by modifying pre-existing variants, adding or deleting the appropriate components or group of components (see Figure 6.8).

It is interesting to notice that this 'degenerated' add-delete approach was not that foolish. In fact, facilitated by the evolution of production planning and control systems, modern software makes the editing and copying of bills of materials easier and quicker than in the past. In addition, the low cost of information processing and data storage favours the proliferation of bills of materials.

However, this particular practice presents a number of problems. Manual handling of the bill of materials may lead to different types of mistakes. Firstly, incompatible options may be included in a bill of materials, since the copy-and-edit process is error-prone. Secondly, BOMs elaborated by different people, at different times, may follow dissimilar criteria. For example, an item, which appeared at level 7 in a

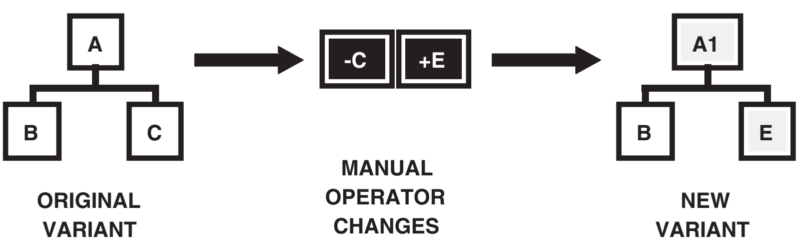


Figure 6.8: Elaboration of a bill of materials after contract with the customer

1997 bill, might mysteriously appear at level 4 in a bill of materials created a few years later. Finally, sometimes creating a new BOM can be easier than retrieving the BOM corresponding to what a customer is asking for. That is why, on many occasions, BOMs are duplicated in the MPCS database.

In addition, although creating a new BOM by copying-and-editing an old one would take about 15–60 minutes, the time required to have the new BOM available is much longer. In fact, the creation of the BOM is done manually by the technical office in batches, e.g. every two days. Consequently, the throughput time to get a new bill is significantly longer than the mere time required to perform the copy-and-edit operation. Ultimately, the elapsed time from customer contact to delivery is negatively affected by the 'degenerated' add-delete approach.

Case 6.3 Add-delete approach for the configuration of electric motors

Rotamax is a company specialized in manufacturing and selling low and medium voltage electric motors, DC motors and synchronous generators. It is a large company, a leader in the sector and part of an international industrial group. Rotamax has enlarged its product offer and is able to customize customer requirements. One of the most important and configurable families among all the products offered by the company, is the three-phase asynchronous electric motor family. These motors may have many different attributes: series and type, polarity, frequency, output, assembling, input voltage, type of variant. The elaboration of the bills of material for the different family variants is based on the so-called 'add-delete logic'. Consequently, if a bill of materials that contains some mistakes is copied, these mistakes are transferred automatically to other bills, creating a spreading effect called a 'spider web'.

Another typical disadvantage is the incompleteness of the BOM due to a lack of updating after substituting some components. The analysis of a particular type of motor, 'Totally Enclosed Fan Cooled' or TEFC, showed that among the 275 variants present in the system, more than 20 per cent of them had errors in the BOM. These errors ranged from 1.45 per cent in the kit of screws to 10 per cent in the connectors.

6.5 Technical model and generic bills of material

The previous paragraphs showed how structured approaches as well as non-structured ones, typically proposed and used for the documentation of product variants, are not adequate for a company that offers configured products. The system of add-delete bills, in fact, is useful only when it is possible to define a 'base model'. When it is not possible to identify a single base model or when there are too many models, the application of this method becomes difficult, due to the complexity

Table 6.1: Analysis of errors in a bill of materials for TEFC electric motors

Component	*#BOM with the error*	*% error on BOMs*	*Component*	*#BOM with the error*	*% error on BOMs*
Terminal	4	1.45	M6 bolt	4	1.45
Connector	26	9.45	Copper washer 6.4 12.5	4	1.45
Lead contact Pg 29	3	1.09	Screw M6 16	4	1.45
Brass lock nut	1	0.36	Elastic washer 8.4	9	3.27
Dutral gasket	3	1.09	Screw M8	2	0.73
Plastic plug Pg 29	8	2.9	Copper washer	26	9.45
Plastic plug Pg 36	8	2.9	Technical document	6	2.18
Plug 39 plastic	11	4	Paint	5	1.82
Plug 49 plastic	11	4	Catalyst	5	1.82
Screw M6 25	8	2.9	Instructions	36	13.1
Copper washer	39	14.2	Plug 42	18	6.54
M6 bolt	4	1.45	Terminal connection diagram	4	1.45
Rosetta copper 6.4 12.5	4	1.45	Screw M8 12	23	8.36
Screw M6 16	4	1.45	Elastic washer M8	23	8.36

of the different links within the add-delete bill. The approach of modular bills is effective only in the infrequent cases of modular products or of very simple assembly/fabrication cycles. The method that we defined as 'degenerated' add-delete bills tends to create errors and delays as well as redundancy in the elaboration of the BOM.

Considering the conceptual scheme of product configurator architecture (Chapter 4), it is evident that the technical model should generate bills of product variants:

- regardless of the particular architectural characteristics of the product (modular, parametric, etc.);
- respecting the hierarchy between components fabricated and/or assembled in steps that follow the production process, that must be controlled separately;
- respecting all the constraints necessary to assure the technical and commercial feasibility of the product.

Generic or virtual bills of material are the instruments used by modern product configurators. The description of this type of bills illustrates their logical function, rather than the details about the way in which these bills are implemented in a computing system, details that change from one application to another.

The main idea that supports the generic bill of materials, is the abstract representation of a set of product variants as the result of the combination of different types of components. Returning to the example of the truck (see case 5.5) we notice that each variant can be described as the combination of a cabin (long, short, high, low), an engine (250 Hp, 300 Hp, etc.), a gear system (manual, semi-automatic, etc.), a type of chassis, wheel configuration, etc. Two variants may differ only for the engine or, in contrast, for all the principal components assembled. All the variants share the types of component necessary and the structure according to which such types are assembled: the engine is not assembled on the cabin, but on the chassis, the gear system is connected to the engine, not to the cabin. Therefore, it is possible to describe a generic truck composed of a cabin, a chassis, an engine, etc. In this way we have defined the first two levels (level 0 and level 1) of the generic bill of materials, in which a certain product, for example the truck, is formed by a number of generic elements, the so-called 'generic offspring': in our case, the cabin, the engine, etc.

These two first levels of the generic bill are enough to understand the logic according to which any variant of the family is obtained (see

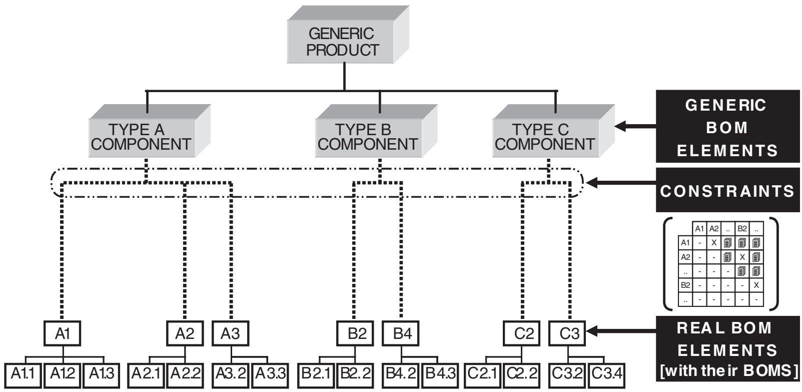


Figure 6.9: Generic bill of materials

Figure 6.9). Since the technical model is the base for generating real bills, it is necessary to indicate in the generic bill, for each 'type of component' or 'generic element of the bill', the 'real components' or 'real elements of the bill' that may be used in the production of the different product variants. For example, the cabin can be offered in four variants (high, low, short, long). This means that under the virtual element 'cabin' there will be at least four 'real' bills of material, each one referring to one of the four available versions (see Figure 6.9).

6.5.1 Constraints

As seen in section 5.5, the components of a truck are, in some way, interdependent. For example, the high and long cabin, particularly comfortable, is a suitable solution if the same driver uses the truck for long periods, as in the case of international long-distance vehicles. In contrast, in the case of a construction vehicle, the super-comfortable cabin is not really necessary, but it is important to have particularly strong suspension, for example, a leaf spring system. Consequently, it is necessary to determine a number of constraints between real elements that may lead to different virtual variants. In the case of Figure 6.9, A2 might not be combined with B4, and A1, B2 and C2 might be incompatible components.

It is important to remember that many constraints *can be expressed within the commercial model* (for example, after selecting the version 'quarry/construction truck' it is not possible to modify the option 'suspension' or to select an engine with lower power). Doing so, we eliminate the necessity of considering incompatibilities between components in the technical model. There is another type of constraint called an *inserting constraint* that will be discussed in section 6.8, linking commercial model and technical model.

6.5.2 Multi-level models

Let us consider again the example of the truck. It is reasonable to think that in the real world there are more types of cabins. First of all, the commercial characteristics of the cabin are not limited to its dimensions: the customer can specify many other attributes, such as the interior upholstery (synthetic-leather, grey fabric, black fabric, etc.), heated windows, type of air-conditioning (there are three or four available variants), stereo (CD player, two, four or six loudspeakers, etc.). Secondly, many components change due to other options. For example, the type of gearbox modifies the gear control in the cabin; an extra set of lights requires a specific switch on the dashboard, etc. The number of dashboard variants for a family of road trucks can amount to a thousand options! It is clear that we may have thousands of variants just for the cabin, and this will determine an excessive proliferation of bills of material. For these reasons, it is necessary to reduce the logic of generic elements to a lower level. In other words, we cannot say that the product is modelled simply by combining generic elements that correspond to real elements. Whenever it is possible, to a generic element at level 1, for example the cabin, may correspond generic elements at level 2, for example the dashboard, and so on and so forth (see Figure 6.10). Obviously, possible constraints may be specified even for the real offspring of a generic element at a level lower than the first one.

6.5.3 Self-codifying components

So far, we have considered the situation in which product variants are obtained by combining real components associated with generic elements while elaborating the technical model. In other words, we assumed that all the possible components necessary to manufacture a certain product variant had to be known a priori, before receiving customer specifications. But there are many other situations in which this fact is not confirmed. Very often, one or more product components require different measures, calibrations or settings, which are described by parameters which can take infinite values within a given interval. For example, if we consider the case of a company that

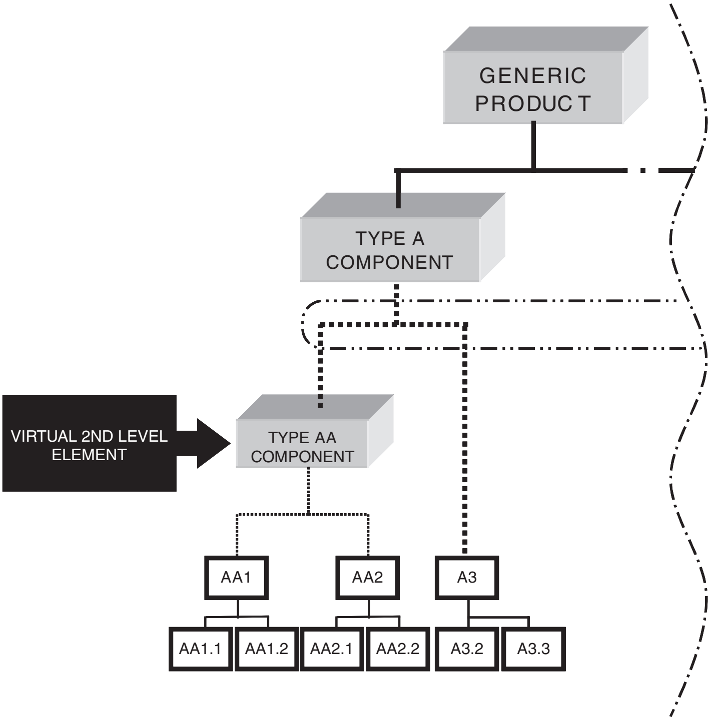


Figure 6.10: Presence of types of components at levels lower than the first one

manufactures made-to-measure kitchen furniture, it is unthinkable that the company will define a priori all the options that the customer may specify, which in any case would be a waste of time. On the contrary, the system must be able to set the dimensions for the parts that constitute the piece of furniture that will be 'made to-measure'. The product model and, consequently, the generic bill of materials must be able to create new components, based on customer requirements, as long as such components are within the field of admissible variations (most probably the company would refuse to manufacture a kitchen table thirty metres long!) We are policode, in this case, of *self-codifying generic elements* (see Figure 6.11).

A self-codifying generic element will necessarily contain a *rule* capable of defining a new real element, based on the parameters specified in the sales dialogue. This rule will codify the new element and will set the appropriate fields in the master item file.

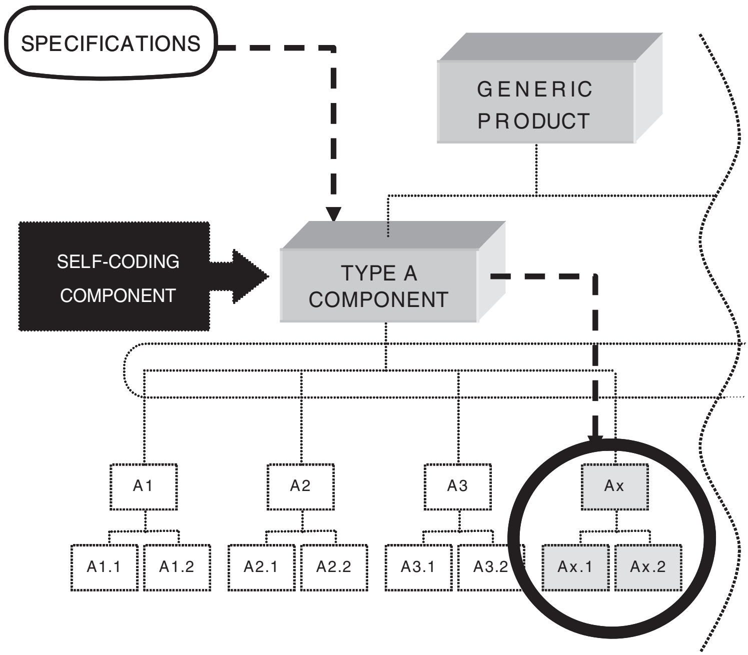


Figure 6.11: Dynamic generation of new real elements in a bill of materials

Case 6.4 Generic bills of materials for a normalized mould-base

Straziametallo is a small company (forty employees) that manufactures normalized mould-bases. A mould-base is needed to connect die to press and is essentially made of a stack of steel plates that are connected by means of guide pillars that slide into holes drilled in the plates. The plates can be made of different types of material, have different sizes, have different hole patterns and be drilled with different hole-diameters. The product presents two problems: the generation of variants by combining standard elements, and the generation of non-standard elements (plates). The product model elaborated by the company consists in, at level 1, identifying the plates (as well as the guides, pins and other components) that can be combined within a certain product family. Regarding the different plates, the system works as shown in Figure 6.12.

For each plate, the configurator searches among the already manufactured plates, in order to find one that is compatible with the customer's specifications (see no. 2 Figure 6.12). If such a plate is not available, there is a configuration rule of finished plates (see no. 3) that searches for a compatible raw plate among the ones already made (see no. 4: Raw plate 1, 2, … m). If it exists, the elaboration required is inserted, codifying the new finished plate variant that is added to other finished plates (see no. 5). Whenever the requested raw plate is not present among the ones already made, there is a configuration rule of raw plates

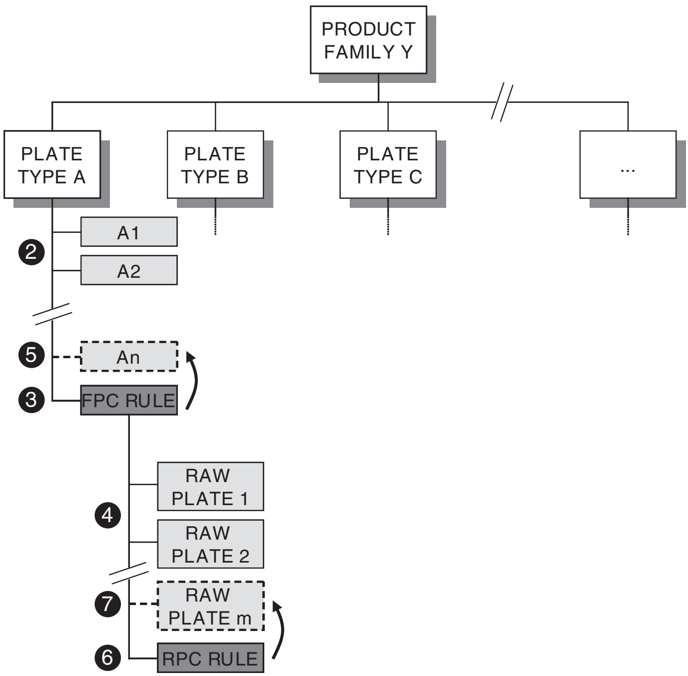


Figure 6.12: Schematic technical model for normalized mould-bases

(see no. 6) that creates a new raw plate, codifies it and adds it to the other raw plates (see no. 7). Then, this plate will be used by the configuration rule of finished plates and consequently a new finished plate will be generated.

6.6 Unification of product views in the technical model

As already mentioned, the technical model must be able to describe the product from the point of view of different departments and, in a general sense, must support different decisions and activities. The single physical description, i.e. the bill of materials, is not always enough to satisfy the needs of different departments. Other information, such as production or assembly cycles, may be influenced by customer specifications, and in this case they will be included in the generic bill. The availability of a complete cycle may be necessary to schedule production, to establish a reliable delivery time, to determine product price or to control margins. Production, sales and cost accounting may be interested in having such datum as output of the technical configuration. Similarly, especially in the case of durable goods, the information included in the manual for the use and maintenance of the product may depend on the characteristics requested by the customer, so it is necessary to consider the 'configuration' of the manual itself. The generic bill offers the opportunity of integrating this information, in order to have a single model that, based on customer specifications, generates bills of materials, cycles, manuals and whatever may be useful for the company. However, these opportunities may be limited by the capability of the selected product configurator (see Chapter 9).

As shown in Figure 6.13, the logic used for the elaboration of the different product viewpoints is the same as the one used for the bill of materials, already described. Therefore, even different production cycles can be conditioned by constraints. For instance, the assembly cycle 'Cycle A+B' shown in Figure 6.13 may vary if we select the real element A1 or the real element A2. The parts of the cycle 'Cycle A1', 'Cycle A2' and 'Cycle B1', however, that are included in real elements, will be considered as invariable. The same logic is applied to parts of the manual 'Mu A+B', 'Mu A' and 'Mu B'.

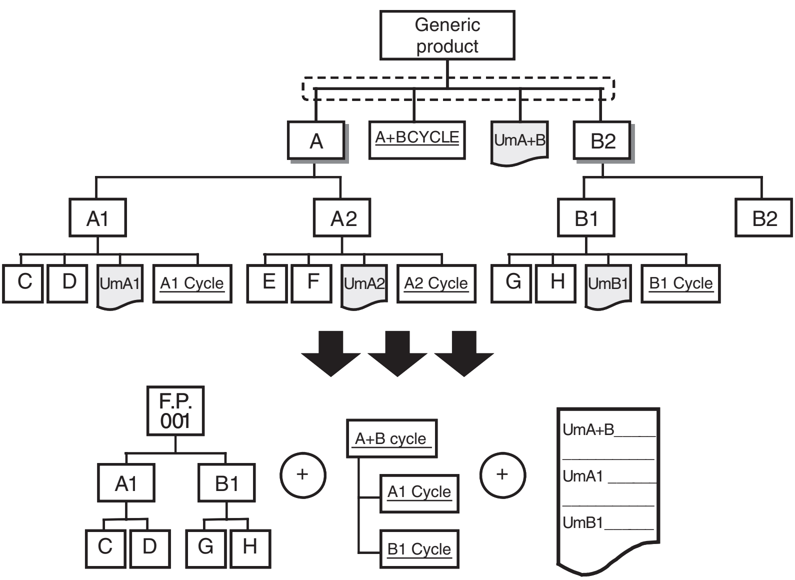


Figure 6.13: Types of information present in the generic bill

Case 6.5 Silentwhistle technical model

Let us consider once again the example of the pressure regulators manufactured by Silentwhistle, mentioned in section 5.3. The analysis of the configuration process of the regulator showed that configuration concerns not only product components, but also the production process and the product-related service. Regarding the production process, even if the assembly took place following the same set of operations for all the variants, some aspects – such as the varnishing and the selection of a suitable gauging spring – were not made following the bill of materials, so the operator needed specific instructions. As for product-related services, the customer may have specified in the order some options – such as packaging, kit with spare springs, customized or certified technical drawings – that required some activities in the technical office or in the warehouse, specifically related to his order. In other words, the specifications included in the order pertained to different departments.

Some specifications, related to the product, concerned the technical office. Other specifications, related to the process, concerned the shop floor. Finally, service specifications concerned the warehouse and, again, the technical office. Figure 6.14 shows the information present at level 1 in the original bill of materials (in black) and the added information necessary to satisfy the requirements of the different inter-company sections involved in the configuration process (in grey: process information; in white: service information).

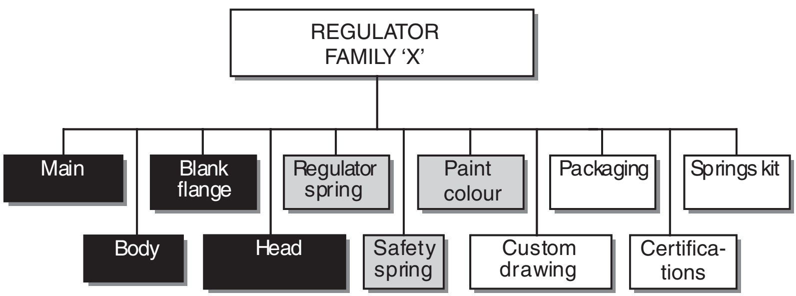


Figure 6.14: Proposal of generic bill of materials for Silentwhistle

6.7 Criteria for defining the technical model

Using a technical model based on the concept of generic bill of materials, basically, means following a series of rules in the elaboration of a formal description of a product family. The company must use the logical instruments that support a generic bill of materials so as to describe the product in the most appropriate way.

It is necessary to consider two facts. Firstly, the description must be as simple as possible, in order to facilitate the adoption of a customization approach based on product configuration. Secondly, the resulting product model must be easy enough to be maintained, that is to say that the updating and possible extension of the technical model must not represent insurmountable obstacles. The reader should be warned that these are challenging goals to meet. However, an appropriate description of the product and consequently, a correct structure of the technical model, are fundamental in getting the maximum from a product configurator.

Considering the fundamental criteria to properly design the product model, it is important, above all, to understand the appropriate number of levels in the generic bill of materials. We may limit the definition of generic elements at level 1 or we may define generic elements at higher levels. The 'optimal' decision depends largely on (1) the number of commercial attributes that persist on a certain element (various, in the case of a truck dashboard) and (2) the proliferation of real elements due to the presence of links with other generic elements of the bill of materials (again, the case of the truck dashboard). When these conditions are present, to configure only at level 1 will cause an enormous proliferation of the corresponding real elements, with many redundancies, and in the case of condition (2) it will cause complex links between variants of generic elements. On the other hand, if there are only a few variants under a certain generic element (for example the different Swatch watchcases) or the real elements under a certain generic element are numerous, but are not connected to the others (for example their straps), it is convenient to limit the configuration to level 1, i.e. only to generic elements at level 1.

Another important consideration, regarding not the depth but the width of the generic bill of materials, concerns the definition of generic

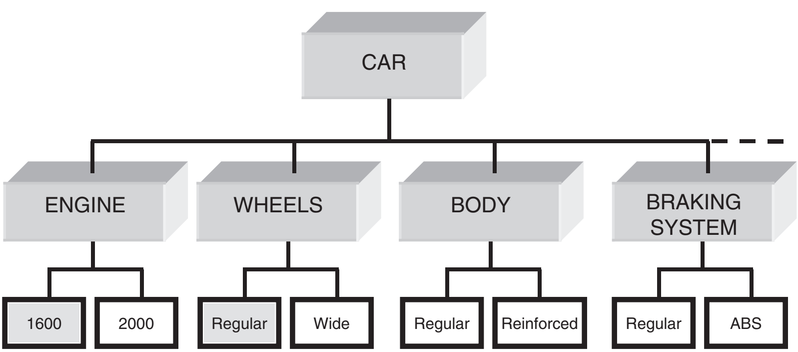


Figure 6.15: Example of a description of technical information focused on physical subsets

elements. The same product can be divided into different 'portions' according to the criterion adopted. Let us take, for example an automobile. A possible division of the structure of an automobile, in configuring the product, may be based on its physical components (see Figure 6.15): engine (1600 or 2000cc), wheels (standard or wide), body (standard or shock-resistant), brake system (standard or with ABS), etc.

The description of the automobile may also be represented as a combination of real elements associated with generic performances such as sports characteristics (traditional vs. sports car), safety (standard vs. reinforced), etc. (see Figure 6.16).

It is interesting to notice that although we are considering the same automobile, the real elements of the bill of materials, present in both cases, are different because they belong to different sets of components. For example, the 1600cc engine and standard wheels, being respectively real elements under the generic group 'engine' and 'wheels', in the bill focused on physical components, belong to the same real element, under the generic group 'sports characteristic', in the bill focused on performances. Moreover, both representations of the same product (considering the example mentioned) are free from redundancies. From a physical point of view, both representations correspond to two different divisions of the same product.

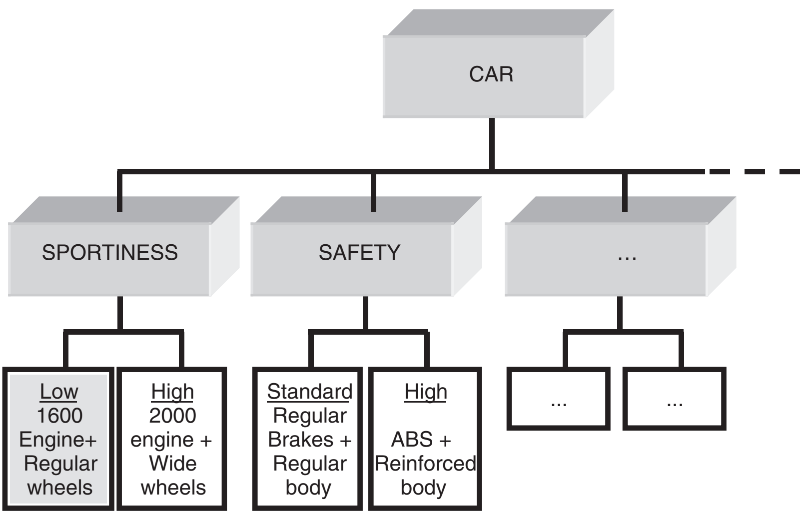


Figure 6.16: Example of a description of technical information focused on performances

Which is the best solution? We must take into account the advantages and disadvantages of both possibilities. It is obvious that the first representation, focused on physical sub-sets, is understood better by a technician, and so, with the first representation, the elaboration of the technical model and its maintenance may be simpler than with other criteria. On the other hand, the second possibility, focused on abstract functions such as safety or on performances, may not be intuitively understood by a technician. Furthermore, the former is advantageous when a certain function or performance is obtained through a subsystem, rather than through a single sub-set: in other words, through a number of components distributed in different parts of the product structure. From this point of view, the specification ABS given by the customer may be related to a real element of the bill of materials formed by components distributed all over the vehicle.

But there is another factor that may determine the most appropriate product description, connected to the fact that, in order to support all the configuration process, the specifications gathered in the sales dialogue must be mapped into the technical model to determine the technical configuration, i.e. it is necessary to connect the commercial model to the technical model.

6.8 Links with sales dialogue

For those product configurators that do not foresee a commercial model different from the technical model, the so-called 'technical configurators', the problem of the links between technical and commercial model is not present, because the latter does not exist. In this case, each question asked of the user corresponds to the specification of one or more parameters directly necessary to select a certain real element of the bill of materials. On the contrary, if the commercial model and the technical model are two separate entities, it is necessary to define a priori how the possible answers supplied by the user contribute to clearly determine which real element, among the 'n' options, must be selected or generated. In other words, it is necessary to indicate how the answers given in the sales dialogue are graphically represented in the technical model. Such representation is made with the support of a particular type of linkage, that we may call an *inserting constraint*. Let us consider the example shown in Figure 6.17. For the answers to the question 2 (ans. #2a, ans. #2b and ans. #2c) the inserting constraints simply specify which real element of 'component type a' must be selected in the elaboration of the bill of materials for

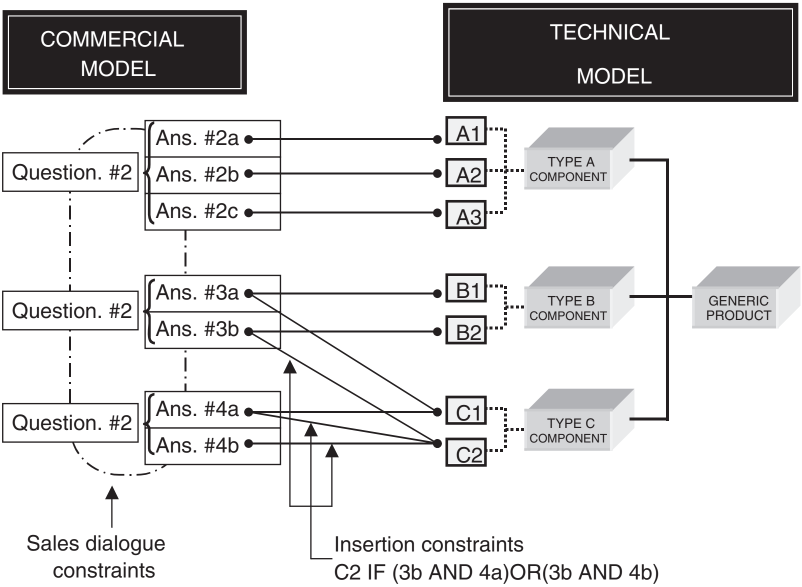


Figure 6.17: Sales dialogue-technical model graphic representation

this variant: respectively A1, A2 and A3. Regarding the real elements C1 and C2 of 'component type c', instead, each of them corresponds to a combination of answers. C2, for example, depends contemporarily on ans. #3a, ans. #4a and ans. #4b. The constraint, in this case, must also indicate the nature of the dependence. In most cases, this result is obtained by means of a Boolean expression, as shown in the figure.

Returning to the problem of the 'optimal' representation of the product structure, a crucial aspect is whether the commercial description of the product is close or far from the description of the product in terms of physical components. When the sales dialogue describes the product in terms of physical components, mapping the questions of the sales dialogue into physical components is a straightforward task. For example, if you are asked what kind of hard drive you want in your PC, of course, the given answers (50GB, 100GB, etc.) will map each one in a specific real component within the technical model (50GB HD, 100GB HD, etc.). When the sales dialogue describes the product in term of its use, rather than its physical components, mapping the questions (within the sales dialogue) to real components (within the technical model) may become a cumbersome task. For example, if you are asked whether you are going to use your PC for text editing or photo editing, many components will be affected by this apparently innocent question: hard disk, video controller, RAM video, processor speed, screen, etc.

The example of the description focused on the functions and the one focused on the components (see section 6.7) is an 'extreme' case, intended to demonstrate the simple fact that there is not only one possible technical model for a certain product, but it can be made in different ways. In reality, not *all* the possible alternatives are meaningful for a company, but *some* of them are always possible. Clearly, when the company has an articulated and structured production process, which needs bills of materials with a structure reflecting that of the production process (production bills), only some alternatives in the definition of the technical model will be feasible. For example, let us consider the case of a company that makes a product by manufacturing three complex sub-assemblies, A, B and C, in different departments, and then integrates them along a final assembly line. In this context, the production bill should mirror the manufacturing and assembly process, i.e. it should include A, B and C as distinct elements at level 1. On the other hand, when the production process is simple, the production bills are less useful, with higher degrees of freedom in the definition of the technical model. For example, if, in a company that assembles components, the workers use a simple part list to accomplish their tasks, it is clear that the production bill will not be necessary, and consequently there will be great flexibility in the elaboration of the technical model.

To sum up, in the elaboration of the technical model it is important to have a clear idea of the extreme alternatives of the product representation: the one based on functions/performances versus the one based on physical components. Secondly, it is useful to understand which compromise is appropriate between these alternatives – the one that is the optimal trade-off between (a) the general effort of creation and maintenance of generic bills of materials, and (b) the creation and maintenance of inserting constraints.

7
Other Product Models

Not all the information necessary for the configuration process is expressed through a technical model based on a generic bill of materials. Some fundamental outputs of a configuration system are actually based on logical structures different from the technical and/or commercial model. Other outputs are obtained through logical structures that complement the technical and/or commercial model. All these structures have to be considered as fundamental elements of the configuration system, since they support some of the important subtasks of the system.

This chapter describes the models the configurator uses to:

- codify the product;
- determine product cost;
- determine product price;
- graphically describe the product.

7.1 Codes

A company that offers customized products must continuously codify newly finished products and new components. Hundreds of new codes might be generated every year, and sometimes even thousands. This is a burdensome process since it is repetitive, frequent and error-prone, and it obviously absorbs considerable resources.

Since the early 1960s, coding has been regarded as a fundamental aspect in the organization of companies that face the problem of product proliferation. There are three typical alternatives: the *policode code*, the *hierarchical code* and the *semi-policode code*. The policode code is a sequence of characters where a certain meaning is associated with each position (Figure 7.1). For example, we can codify an automobile with the three first digits indicating the family (Alfa Romeo '156' rather than '147'). Another two digits indicate the capacity ('20' for 2000cc, '16' for 1600cc, etc.). A fifth digit may indicate the colour and the interior ('A' for alcantara, 'N' for black leather, etc.). The advantage of a policode code is that it allows us to understand immediately what product it refers to, and at the same time it can be rapidly generated. Yet, policode codes necessarily tend to be too long, and, besides the limitation of ERP software, over-long codes are impractical. Moreover, when a product has an enormous quantity of variable attributes, a policode code becomes difficult to use. Even when we use an alphanumeric code, where each digit can take up to 35 different values, very frequently we may run out of characters at our disposal because of the proliferation of options. Let us assume, for example, that the Alfa offers 40 interior variants. It will not be possible to indicate this attribute with only one digit, 40 – 35 = 5 interior variants. An alternative solution to this problem is to interpret the value of the digits devoted to describe 'interiors' together with the value of some other digit. Suppose we have only one digit to identify the interior variants: '1' = black, '2' = light grey ... and 'Z' = deep blue (35th interior option). When we get to the 36th interior variant, i.e. 'Sahara beige', we are not able to codify a car variant with such an interior option, since we have only 35 different letters and numbers. To solve this problem without adding digits to our code (something you cannot do overnight), we can attribute different meaning to the 'interiors' digit, based on the values of another

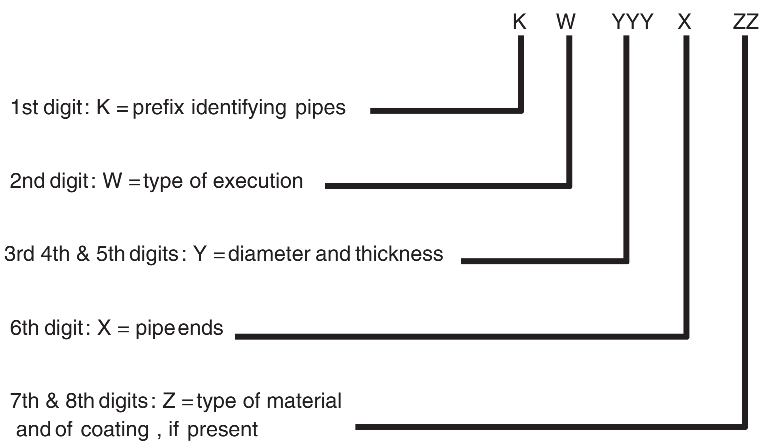


Figure 7.1: Polycode

digit, for example the 'fuel type' digit. Suppose the 'fuel type' digit can have two values 'D' = diesel and 'B' = gasoline. We might allow the 'fuel type' digit to take other two values 'K' (diesel option) and 'J' (gasoline option). Now, we can stipulate that if the 'interior' digit has the value '1' and the 'fuel type' has the value 'K' or 'J', the interior variant we are referring to is 'Sahara beige'. Alternatively, if the 'interior' digit has the value '1' and the 'fuel type' has the value 'D' or 'B', the interior variant we are referring to is 'black'. When these modifications overlap through time it is always more difficult to read the code 'at a glance', and there is a considerable increase in mistakes in the generation and reading of codes.

The logical opposite alternative to the policode code is the hierarchical code, where the meaning of each symbol depends on that of the preceding symbol (Figure 7.2). For example, we can codify a ball bearing using the first digit for the class, the second for the sub-class, the third for the group and the fourth for the series. Unlike policode codes, hierarchic codes are very compact. Yet, they foresee an orderly, hierarchic description of the product family that is not always possible to generate or maintain.

A third alternative, the *semi-policode*, very frequently used, combines characteristics of both types of code with non-structured approaches, where new product variants or new component variants are simply codified by adding a unit to a numerical section of the code. So, after a part of the code that follows a policode or hierarchical logic, there is a second part which we define as 'version counter', which indicates the number of different product variants with an identical policode or hierarchical part. Returning to the example of the automobile, we may have 1 + 1 policode digit (model + body colour) and leave two additional numeric digits for the combinations of five

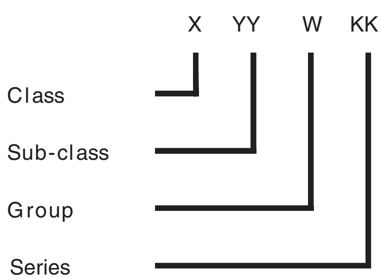


Figure 7.2: Hierarchical code

optional components: air conditioning, ASR, radio, fog lights, heated seat. Suppose we have three models (A, B and C) and four colours (B = Black, W = White, O = Orange, G = Green). The first time we code a model A car, with White body, we have the product code AW01, no matter what options have been selected. For example, we may have air conditioning and ASR, but not radio, fog lights and heated seat. The next time a model A car, with White body, has to be coded (evidently because the first time some customer orders a certain mix of options), the code will be AW02, and another mix of options will be associated with such a code. It is important to note that the code BG01, for example, will not necessarily have the same combination of options as AW01. BG01, in fact, simply indicates the first B model car with Green body that has been coded.

There are many other types of codes (vertical, horizontal, decimal, alphabetic, etc.). Our purpose is to give an overall vision of this subject. The reader will be able to find further information in specialized texts.

Product configuration systems do not modify the criteria according to which product components and finished products are codified. To automate the creation of a code requires a precise definition of the criteria used to generate it. In other words it *imposes the definition of a robust coding plan*. From the technical point of view, the transfer of the code rule from the technician to the product configurator takes place by means of specific structures, called *coding engines*, which require the definition of the criteria used to codify the product or the component. In order to define a code rule, it is necessary to determine which answers of the commercial model influence the code. In the case of a code for finished products, we may consider all the answers supplied by the salesman, but this is not necessarily true. In fact, some of the questions asked in the sales dialogue may concern, for example, packing, kind of delivery and the presence of particular product documentation. But none of these is considered in the product code. Therefore, the first step is to evaluate which questions, within the commercial model, have an influence on the product or component code.

The second step will be the definition of the appropriate length of the code. It is necessary to remember that the length is usually constrained by company information systems and subjected to management approval. However, to change the coding plan when the product configurator is implemented could be a possible option.

The third step is the definition of the code structure. A typical structure found in many product configurators is made up by a root + a policode part + version counter (see Figure 7.3).

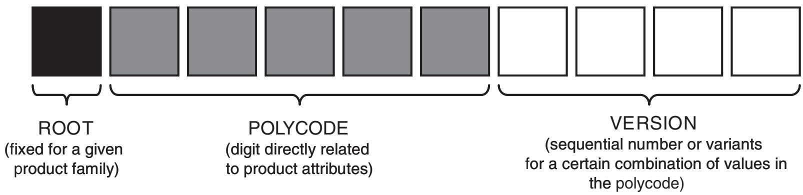


Figure 7.3: Logical structure for code generation

It is evident that the final step is the definition of the association of the policode fields with the answers of the sales dialogue, as shown in Figure 7.4. There are some questions (e.g. #4 and #5) that are not

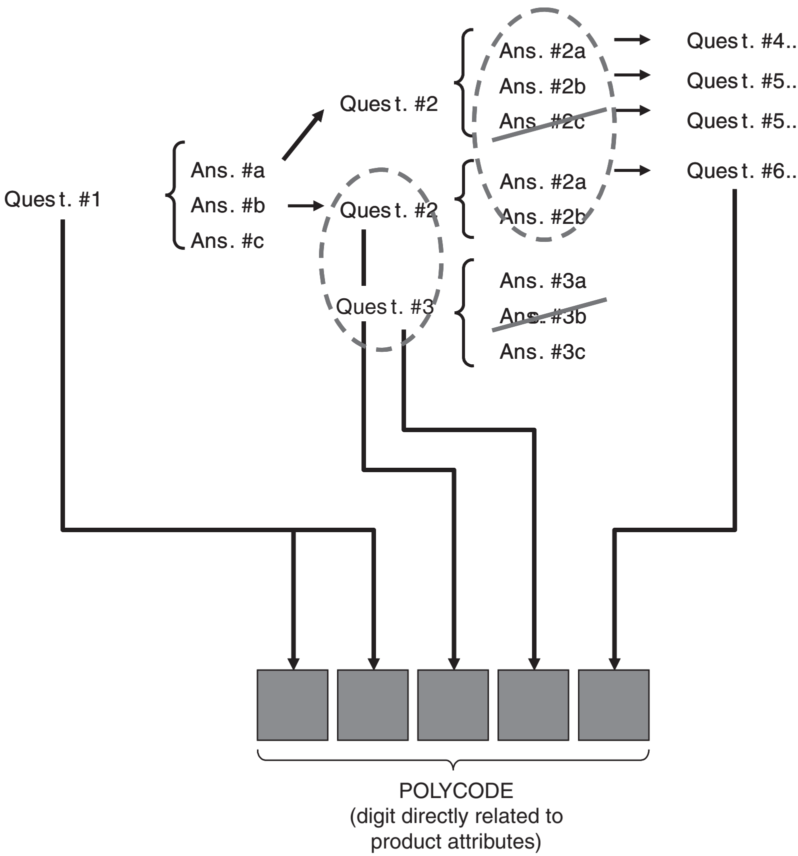


Figure 7.4: Generation of the polycode in the product code

associated with any field of the policode part of the code. This means that the answers to these questions will be gathered by the version counter. For example, if it is requested to make a configuration that, regarding the questions #1, #2, #3 and #6, is identical to a configuration already made (with the same policode part), the system will compare the answers in the old configuration to those in the new one. If there are no differences, the old configuration will be retrieved and it is not necessary to create a new one. If this is not the case, the system will check how many configurations have already been made – with the same policode part of the configuration we are defining – and will assign it with a new version counter.

It is important to notice that, when the configuration is made with remote devices, for example with portable computers used by the sales force, the mechanism of the version counter introduces some difficulty. Various salesmen may create independently different product variants, with the same policode part and different version counter. In this case, it is necessary to introduce a further procedure for the approval of codes, to avoid the problem. Yet, as wireless data communication is becoming widespread and cheap, this difficulty will be less important in the future.

7.2 Model used to determine cost

To determine the cost of a product that, due to its numerous customizable attributes, can be manufactured using many different configurations represents a laborious task. Some attributes may influence the finished product cost in different ways, for example the processor of a personal computer compared to the keyboard. Moreover, the same attribute may have different costs as in the case of an automobile: a conventional headlight rather than a xenon headlight. Sometimes apparently irrelevant attributes, for instance special designs for a certain technological product, may require considerable resources.

The uncertainty about product cost is consequently translated into the uncertainty about the appropriate price of the product configuration. If we do not know how a customer's choices influence product cost, we are not able to estimate correctly the price in the sales transaction. The company runs the risk of being underpaid, or of asking too much for the customization. In general, the lack of precise information about configuration costs makes an accurate management of the margins impossible.

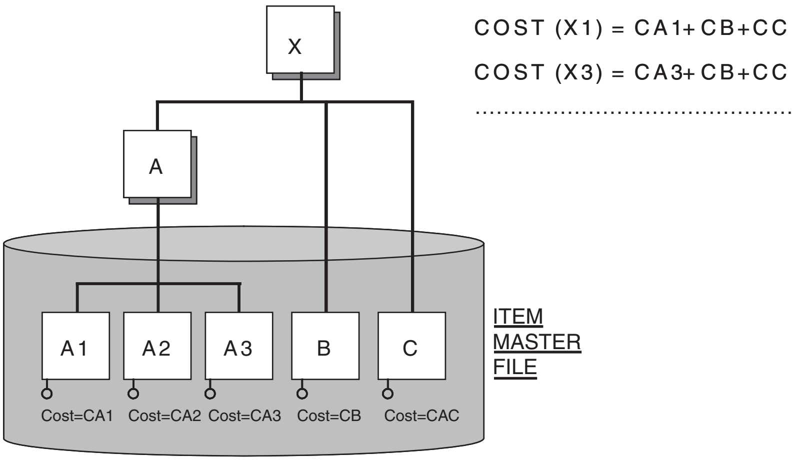


Figure 7.5: Product cost obtained by 'summing up' the cost included in the bill of materials

The costing of the materials and elaborations necessary to obtain the product can be estimated using the generic bill of materials, which constitutes the technical model. The information regarding the costs of the different components is in the detailed description of the parts, that gives origin to the configurator. Evidently, the cost of the product variant can be estimated by the ‘summing-up’ of the costs of real components included in the bill of materials, obtained instantiating the virtual model (see Figure 7.5).

In the majority of cases, there is also a cost related to assembly. When this process is independent from the product characteristics, such a cost may be simply considered similar to a predetermined cost that is added to the components costs. But, very frequently the production or assembly cycle may change from one configuration to another, with considerable consequences for the industrial costs necessary to support the elaboration of the product variant. The technical model structure, i.e. the generic bill of materials, is useful to evaluate the cost of the product variant including also the costs of the production cycle. In fact, as mentioned in section 6.6, the generic bill can also include parts of the production cycle. As in the case of components, a predetermined cost can be assigned to some parts of the cycle. Following the same logic applied to components, the overall cost of a product can be estimated by the ‘summing up’ of not only components costs, but also of the predetermined cost of cycle elements.

The costs of cycle elements are not always predetermined, for example, in the case of *parametric components*. The elaboration of such components generally requires the modification of a certain activity that takes place in the production and/or assembly cycle of the component. For example, to offer customized shelves requires a made-to-measure cut of the boards, i.e. a 'configuration of the production cycle'. This fact may have an influence on product costs. In fact, different values in the parameter that has to be adjusted may imply longer times in the production process, determining changes in the industrial costs that support the elaboration of the component. To determine the cost of the production activities related to a product variant, is not only a question of deciding the static attribute 'production cost' according to the hierarchical logic established in the product model. It may also mean evaluating, dynamically, the attribute 'production cost' according to the answers gathered though the commercial model. In a second moment, the estimated costs will be consolidated using the logic structure of the technical model (see Figure 7.6).

Case 7.1 Determining the cost of a configurable product: the case of office chairs

Nuvola is a medium-sized company that manufactures office chairs and exports more than 50 per cent of its production. Due to the strong competitiveness in the industry and the wide range of products made in its three automated factories, the company needs a very precise control of its costs. Moreover, 20–30 per cent of the offers that the sales office makes annually include variations of the

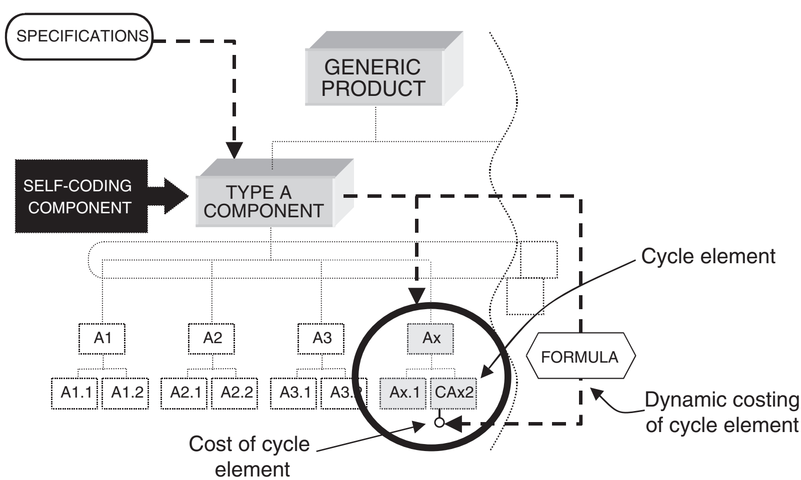
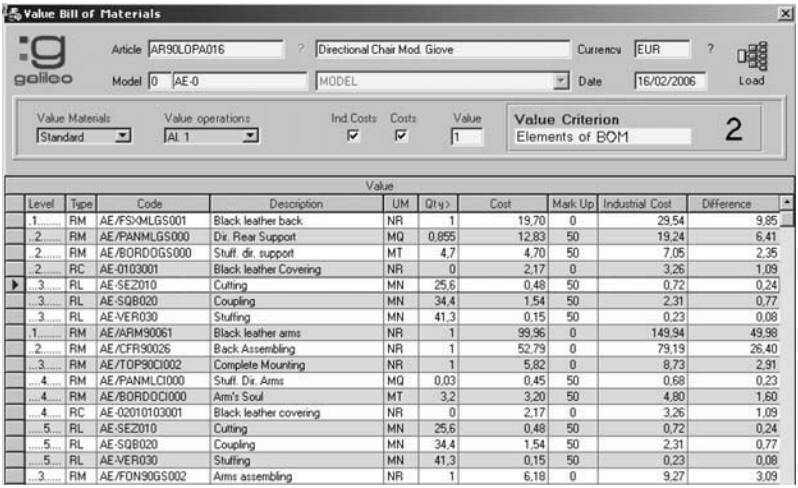


Figure 7.6: Cost of a cycle element: dynamic estimation



Level	Type	Code	Description	UM	Qty>	Cost	Mark Up	Industrial Cost	Difference
.1.......	RM	AE/FSXMLGS001	Black leather back	NR	1	19,70	0	29,54	9,85
..2.......	RM	AE/PANMLGS000	Dir. Rear Support	MQ	0,855	12,83	50	19,24	6,41
..2.......	RM	AE/BORDOGS000	Stuff. dir. support	MT	4,7	4,70	50	7,05	2,35
..2.......	RC	AE-0103001	Black leather Covering	NR	0	2,17	0	3,26	1,09
...3......	RL	AE-SEZ010	Cutting	MN	25,6	0,48	50	0,72	0,24
...3......	RL	AE-SQB020	Coupling	MN	34,4	1,54	50	2,31	0,77
...3......	RL	AE-VER030	Stuffing	MN	41,3	0,15	50	0,23	0,08
.1.......	RM	AE/ARM90061	Black leather arms	NR	1	99,96	0	149,94	49,98
..2.......	RM	AE/CFR90026	Back Assembling	NR	1	52,79	0	79,19	26,40
...3......	RM	AE/TOP90CI002	Complete Mounting	NR	1	5,82	0	8,73	2,91
....4.....	RM	AE/PANMLCI000	Stuff. Dir. Arms	MQ	0,03	0,45	50	0,68	0,23
....4.....	RM	AE/BORDOCI000	Arm's Soul	MT	3,2	3,20	50	4,80	1,60
....4.....	RC	AE-02010103001	Black leather covering	NR	0	2,17	0	3,26	1,09
.....5....	RL	AE-SEZ010	Cutting	MN	25,6	0,48	50	0,72	0,24
.....5....	RL	AE-SQB020	Coupling	MN	34,4	1,54	50	2,31	0,77
.....5....	RL	AE-VER030	Stuffing	MN	41,3	0,15	50	0,23	0,08
...3......	RM	AE/FON90GS002	Arms assembling	NR	1	6,18	0	9,27	3,09

Figure 7.7: Costs included in a real bill of materials (by Sanmarco Informatica)

standard model, reaching a total of 500–800 new product configurations in a year. For these reasons, Nuvola paid great attention to the costing model, implemented in its configurator, taking into account the costs of components as well as production and assembly costs. Figure 7.7 shows the bill of materials used for an armchair.

As we can see, there is a final assembly at level 1, which gathers back-rest and arm-rests (seat, gas bar mechanism, head-rest, base and wheels are indicated on the rest of the screen, which is not shown in the figure). There are other kinds of assembly at lower levels, which include back-rest assembly (at level 2), cutting of stuffing and coupling the stuffing to the back-rest (at level 3), cutting of upholstery and back-rest covering (at level 4).

7.3 Models used for determining product price

To offer customized products presents specific difficulties in the evaluation of the costs of each product variant. In the same way, it presents some problems in the determination of product price. After evaluating the cost of the product variant, as described in the previous paragraphs, it is possible establish the price, applying an appropriate mark-up. Depending on the complexity of the product configurator and of the management control system employed by the company, it is possible to use one or more mark-ups. This means, for example, that the production cost of a component that is manufactured in one department of the company may be higher than the cost of a component manufactured in another department. Similarly, some 'exotic' components may be more expensive because they are not manufactured frequently. Alternatively, a higher price may be calculated according to the customer, the geographic area, etc. following typical criteria found in price lists of commercial forms included in management software.

The determination of a price by applying a mark-up on the costs of the bill of materials is not the only logical possibility. In fact, there is a second alternative, represented by the evaluation of the answers supplied by the customer during the sales dialogue. Especially when the sales dialogue is not abstract, i.e. when it is focused on the physical components rather than on the functions, it is possible to build up the price, step by step, reading the customer's answers that describe the product. In practical terms, this means that each possible answer in the commercial model is associated with a price. When the customer completes the sales dialogue, the prices associated with the customer's choices are added up, and in this way when the dialogue is finished the price related to the configuration selected is also available (see Figure 7.8).

This approach presents some advantages. Firstly, it is a quick solution, which does not require the continuous updating of prices in itemized parts and cycles. Secondly, price updating is relatively simple, since prices are directly linked to the questions asked by the customers

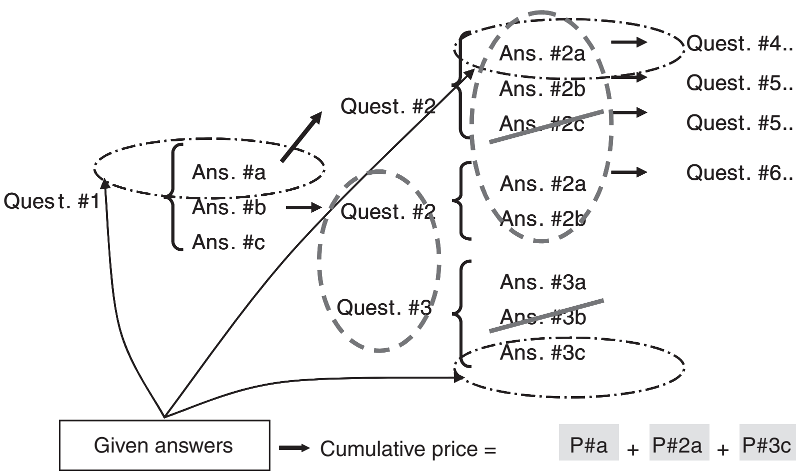


Figure 7.8: Price definition by means of answers gathered in the sales dialogue

during the sales dialogue. Finally, in this way, it is possible to estimate the price of a product that takes into account even some customer requirements that are not necessarily included in the technical model, for example, ways of shipping the product or special certificates, that usually generate costs and therefore affect the price, but are not necessarily present in the virtual bill of materials that supports the technical model (see Figure 7.8).

7.4 Geometric and graphic models

In some situations, the code, price, cost, bill of materials and production cycle resulting from the configuration process are neither optimal nor enough. Very frequently, designs that supply a graphic description of the configuration are requested. The degree of complexity of the representation depends on how the company is willing to use such representation. For example, a company that manufactures customized transformers may be interested in giving its customers a precise idea of how bulky the power transformer is. In the case of customized lifts, the requirements of the graphic output are more articulated, since it is necessary to have a greater amount of geometric parameters, in order to understand the real shape of the variant and its dimensions. Considering a more complex situation, for example, the normalized mould-bases discussed in Chapter 11, it is necessary to have an output of the configuration, a real technical design with graphic displays, measures and other specifications that are needed.

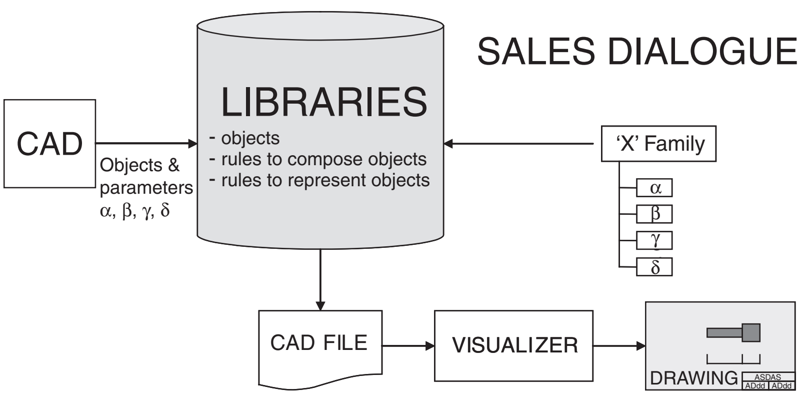


Figure 7.9: Generation of graphic representations

The generation of graphic outputs is based on the concept of 'parametric object', i.e. a graphic object (such as a prism with hexagonal base) with some parameters associated with its dimensions (e.g. side of the base, height of the prism). Considering the values of both parameters 'base' and 'height', it is possible to obtain a certain type of prism, characterized by a certain base and a certain height, and then represent it by means of a graphic system. Figure 7.9 illustrates the logical procedure used to dynamically generate a graphic representation of any product variant.

Most products can be represented as a composition of elementary solids (prisms, cones, spheres, etc.) or of elementary flat surfaces (polygons, circles, lines, etc.). These objects can be generated using a CAD system, and for each of them we can establish the parameters that will be dimensioned during the configuration process (α, β, γ, δ in Figure 7.9). In this way, we get a *library of objects*. To use these objects in the creation of a design, it is necessary to determine the rules according to which these objects are obtained. If we consider a lathed piece, characterized by an α length with a β diameter and then by a γ length with a δ diameter, it will be necessary to indicate that the product is obtained by combining a circular prism, with an α length and a β diameter, with another circular prism, with a γ length and a δ diameter, that must be coaxial. Moreover, some rules for the representation of objects must be defined, such as rules used to write a file (for example a vector file) that can be interpreted by a graphic system (for example Autocad L T). In this way, the user can see the graphic results of the configuration.

It is important to notice that not all the product configurators present on the market are provided with graphic modellers, and that the generation of graphic models adds costs to the process of implementation and maintenance of the system, costs that must be carefully estimated.

Case 7.2 Generation of graphic outputs as a result of configuration: the case of a furniture factory

Summit is a company that produces metal and wooden office tables. Yalta is the leading product line of tables for managerial meetings. These tables can be ordered in different kinds of wood (cherry, walnut, oak) with several incorporated accessories (microphones, electric sockets, headphones jacks, etc.) in different measures, regarding diameter (2 ÷ 4m), height from the ground (900 ÷ 1100mm) and position of the legs (100 ÷ 300mm from the table borders). As shown in Figure 7.10, the table is represented as a composition of five cylinders: the top defined with the parameter '*D*', diameter, and four legs defined with the parameter '*l*' length.

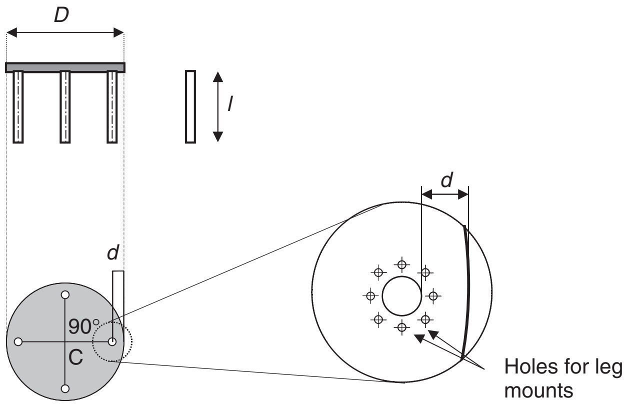


Figure 7.10: Geometric model of a configurable table

The design shows that the construction of the graphic model required the specification of how the legs are arranged with respect to the top: they are orthogonal to the top, with a constant distance between them and equidistant from the centre. Following the customer's specification, the distance of each leg from the edge was assigned the parameter '*d*'. After specifying '*d*', the program can design the scheme of holes on the top, necessary to fix the legs. When the scheme is defined, it is possible to automate the generation of the program that the machining centre will use to drill the holes on the wooden boards.

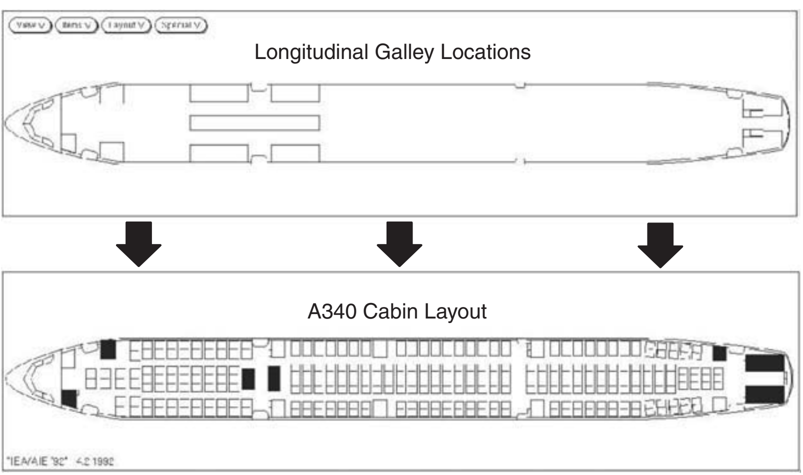


Figure 7.11: Prototype of configurator for A340 cabins

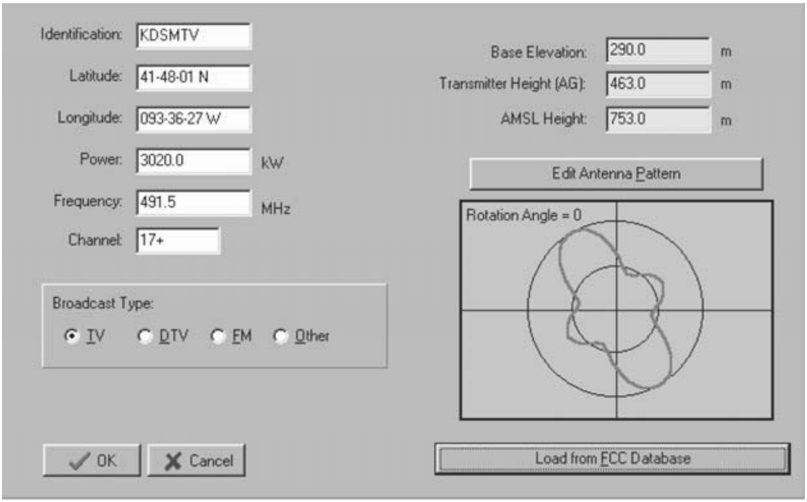


Figure 7.12: Antenna pattern configurator

7.5 Other models

The possible product models, that can be defined to support the automatic generation of different outputs, are virtually unlimited. The cases found in the companies that manufacture configurable products can be absolutely amazing. And also the products to configure and the tasks associated with their configurations may vary from case to case. For example, Figure 7.11 shows the prototype of a configurator that was created to support the A340 Airbus cabin interior fittings. In Figure 7.12, we see the example of an antenna configurator that provides as an output a polar diagram of the antenna pattern.

Similarly, it is possible to build configurators for solving problems of three-dimensional piping layout, for the distribution of machinery, for the elaboration of bank or insurance contracts. The main problem is that developing these special-purpose configurators is quite expensive. Furthermore, the willingness of the customers to use these innovative solutions may be uncertain.

Part III

Selection and Implementation of a Configurator

This page intentionally left blank

8 Configurators and Management Information Systems

The previous chapters analysed the configuration system regardless of its linkages with the overall management information system of a company. Indeed, the product configuration system does not operate independently, as it interacts with specific sub-systems. The present chapter explores the interfaces supporting such interaction, as well as possible overlaps, complementarities and synergies between the configuration system and the larger management information system in which it is embedded. More precisely, the aim of the present chapter is to:

- illustrate the relationship between configurator and Manufacturing Planning and Control System (MPCS);
- illustrate the relationship between configurator and Product Data Management;
- illustrate the relationship between configurator and Customer Relationship Management.

8.1 Configuration system and Manufacturing Planning and Control System

The configurator, as discussed in section 4.1, is an information system supporting product configuration and modelling processes. When considering the opportunity of a configurator it is first of all necessary to have a clear idea of how it affects the order cycle and, from the point of view of computing systems, how it interfaces with the MPCS. The information that it uses and generates is, in fact, exchanged with the MPCS. Figure 8.1 shows a useful scheme that, although synthetic and extremely general, helps to relate the configurator to both the order cycle and the MPCS.

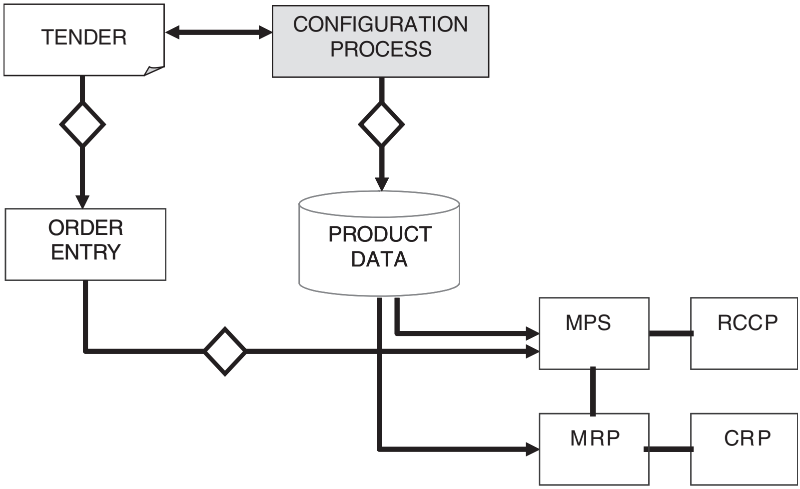


Figure 8.1: Configurator and Manufacturing Planning and Control System

8.1.1 Position of the configurator within the order cycle

In order to understand the link between configurator and MPCS it is necessary to consider how the configurator is inserted within the order cycle. From the very beginning of the process, interaction with the customer is supported by illustrating the product, collecting specific customer needs and identifying the particular product configuration that best fulfils them. The configuration not only specifies the product characteristics, but also allows a precise definition of product cost and, possibly, sales price. Therefore, the product configurator feeds the tendering process with two key pieces of information: product characteristics and price. Once these data are complemented with other information, such as delivery time, payment conditions, etc., the tender can be presented to the customer. If the tender is successful, the data originally generated by the configurator can be transferred to the ERP sales management module, without the need to re-input them – clearly a non-adding value activity. Likewise, the technical information of the product generated by the configurator (BOMs, cycles, etc.) can be transferred to the ERP Manufacturing Planning and Control module(s). To sum up, the configuration system operates before order entry, supporting the generation of the offer in the tendering process, and if it is properly connected to the rest of the management information system, it virtually reduces order entry to a simple confirmation of the offer (see Figure 8.1).

It is important to remember that technical information of the product is transferred from the configurator to the MPCS database only when a new configuration is created. Whenever customer needs correspond to a product variant that the company has already made, the configurator will help identify the product variant, and will not transfer any information to the MPCS database.

On the other hand, the configurator is fed by MPCS data. For example, to determine the cost of a product variant it needs the component cost information stored in such a database. To create a bill of materials, it will pick from the MPCS database one real bill of materials for each of the virtual elements within its generic bill of materials.

The reader should be warned that the MPCS–configurator interface differs from software to software and is highly influenced by the degree of integration between configurator and the company's computing system.

Case 8.1 Linking configurator and MPCS in the ERP: an example

In SAP it is possible to have two steps of configuration: commercial and technical. Both levels of configuration are carried out on the sales order, before the sales order becomes a production order. The Production Planning (PP) module of SAP receives the sales order after configuration and changes it into a production order, on which the configurator does not operate any longer. At this point, modifications can only be 'made by hand' directly onto the BOM, within the production order. This kind of modification does not represent the norm, but an exception to the standard configuration and sales cycle management automatisms.

Once the sales order is changed into a production order and configuration activities are over, control passes to the Materials Management (MM) module. By using MRP, Materials Management module plans all the requirements generated by the sales order and all the corresponding shop and purchasing orders. These orders are released based on identification flags present in the BOM, which indicate whether a specific code (1) must be purchased from an associated company or (2) from an external supplier, (3) whether it refers to materials in stock or (4) if it is internally produced each time it is ordered. According to these alternatives, MRP generates purchase requirements sent to the purchase department, orders of materials sent to the stock management and finally, planned production orders sent to the production department.

Before generating shop and purchasing orders, MRP has to transform the configured BOM into a production BOM, because the logic that supports production is different from that of design and sales, which is used to specify and build the BOM during the configuration process. This manipulation of the BOM is carried out according to two strictly connected procedures: (1) for each level change in the configured BOM, a planning order is created; (2) following a dummy code, the corresponding level change is voided. When the MRP, during the examination of the configured BOM, recognizes a dummy code according to the type of recognition flag present in the identifying materials, it simultaneously controls the types of code in the successive level and carries on reading the BOM, storing the number and position of all the dummy codes identified. When the production BOM is exploited all the levels corresponding to dummy codes are cancelled, and consequently, the real components are shown. The result of such a manipulation process of the configured BOM is a production BOM characterized by a lower number of levels, compared to the initial one.

8.1.2 Determining the delivery date

Most often, an offer cannot be made without specifying a delivery date. In these cases, the configurator ideally should be capable of supplying this information. Different contexts, however, require different precision about delivery times. Sometimes what matters is not defining the earliest possible delivery date, because what customers are really asking for are reliable delivery times. For example, in the case of a wedding dress, you do not need it overnight, but you want to be sure you will get the dress by your wedding day. At other times speed of delivery is more important than delivery reliability. The requirement, therefore, is to compute the earliest possible delivery date. Consider, for example, the case where a custom-made cutting head for a milling machine is broken: the customer will need that part as soon as possible, since the cost of having the machine idle is very high.

Computing the delivery date accurately involves a relatively long sequence of information processing activities. First of all, the bill of material has to be read by the MPCS, and the MRP has to assess the availability of materials. Then, the MPCS scheduler may estimate a completion time, by considering manufacturing resources status and orders portfolio. The delivery date would then be fed back to the configurator, to be finally incorporated in the tender. Of course, the smooth performance of all these information exchanges and computations is eased when the configurator is *embedded into an integrated ERP system*. Were the configurator a *stand-alone package plugged to an ERP system*, we would have to define appropriate configurator–MRP–scheduler interfaces to get our delivery date. Needless to say, this would be costly.

Even in the case of a configurator able to interface with the MPCS to get an accurate delivery date, for both theoretical and practical reasons, such a solution would be of little use. From a theoretical standpoint, we have to consider that there is a time lag between delivery date computation and the confirmation of a specific order, say order A. During such a time interval, other orders may have been entered into the system, say orders B, C and D. If B, C, and D lock manufacturing resources required to fulfil order A, its delivery date will unavoidably slip forward. From a practical standpoint, the solution to getting an accurate delivery date requires running MRP and the scheduler each time an offer is done. Clearly, such a practice would either jam the company MPCS or it would require an excessive investment in information technology.

Most often, the practical solution to defining delivery dates is to use heuristics, i.e. routines that propose a delivery date based on simple rules or computations. Such routines can obtain the necessary information from the MPCS, or follow some simple guidelines, such as proposing a standard delivery time for each product family, adjusting an overall lead time according to seasonality, etc.

8.2 Configuration systems and Product Data Management systems

The configurator, as mentioned before, is fed with product data and elaborates product data, in the form of bills of materials, production cycles, drawings, etc. However, the goal of managing product information in an integrated way is pursued by the so-called Product Data Management (PDM) systems. Therefore, the question is, how do the functions expected from a configurator relate to the functions expected

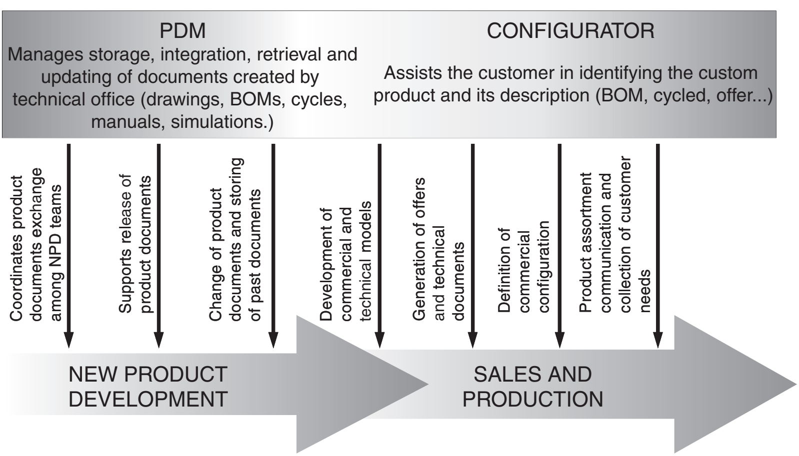


Figure 8.2: Configurator and PDM

from a Product Data Management system? Since there are no clear-cut distinctions between the functionalities of PDM and configuration systems, the present section aims to identify points of overlap, of divergence and areas of possible integration between both systems (see Figure 8.2).

8.2.1 PDM functions

In the 1980s, many advanced companies, while adopting JIT and TQM approaches to boost their operations productivity, realized that they also had to improve product design processes. Unfortunately, the advantages of information technology in product design activities, using tools such as CAD/CAM/CAE were limited by the paper-based document management systems in use at that time in most technical offices.

A crucial point was the engineering–manufacturing interface. Deficiencies were found both in management of engineering change process and in the way technical documents were released to production. Consequently, the first PDM were developed. At the beginning, they were developed in-house by the companies interested in improving their management of technical documentation. Later, specialized software vendors entered the market of PDM applications. These PDM supported the generation and storage of technical documentation released to production and eased the management of engineering changes. With the passing of time, the range of product document types managed by PDM increased: blueprints, diagrams, sketches, schemes, circuits, manuals, lists of materials, production BOM, etc. In their evolution, the scope of PDM systems embraced product data throughout the whole product lifecycle, from concept to disposal.

A second generation of PDM was capable of handling the revision status of product documentation, thus allowing also management of documents in progress. In this way, the goal of the PDM was to support the design team, disciplining the circulation of documents in progress through different departments and people. The capability of supporting the whole design process underlying product development is sometimes added to this software, which is called workflow.

Current PDM systems aim at managing the systematic filing, integration, availability and updating of the documents produced by the technical office (designs, BOM, cycles, manuals, numerical simulations, etc.). Their main functions are to store data in an *electronic vault* and to make it promptly available to users. The system controls the way in which the operators create and modify data, defining the sequence of events that must be verified before allowing any modification. Among the stored data, it is possible to define links or associations that reflect the product structure, supplying to different users the tools and product views that best fit their needs. Finally, the data of the electronic vault can be classified according to a particular physical characteristic or similarity in the production cycle. For example, a designer may need a certain component. Instead of designing it, he may look for a functionally and morphologically similar one across the already designed components. Alternatively, he may pick a similar one and modify it, rather than designing the required component from scratch. This functionality allows the existing components to be re-utilized and standardized and reduces the design activities needed to develop a new product.

8.2.2 Similarities and differences between PDM and product configurators

If we compare what has been said about PDM systems to what has been said about configurators, it is evident that between both tools there are some common functionalities. Nevertheless, the numerous differences between them are enough to confirm that these tools are complementary, not substitutive.

There are some overlapping functionalities because both forms of software have mechanisms that store, create and retrieve product data. Furthermore, in both systems it is possible to classify stored data according to various attributes. Finally, PDMs and configurators permit the definition of the different relationships between the multiple parts of the product, helping to define the bill of materials and associating with each code the documentation necessary to identify and produce a specific product variant.

However, the two systems differ in terms of other functionalities. This is due to the fact that they were created with different aims. PDM systems, in fact, were conceived to manage engineering product data, while configurators were envisaged to gear sales process with engineering and production when custom products are offered. In particular, the main difference between the two tools is that PDM systems mostly do not provide an adequate support to sales product configuration, whereas configurators, due to their essence, do not support workflow management.

8.2.3 Integration between PDM systems and product configurators

The conclusion drawn in the preceding sections brings up a second question. These two complementary tools have common functions and, in part, manage the same data, but how should they be connected? There is no one single answer to this question and many possible solutions have to be taken into consideration, according to the particular characteristics found in the company's context.

The easiest solution, apparently, consists of letting the two systems operate independently. In this case, there must be a complete duplication of the common product data, which have to be stored separately in the vault of the PDM system and in the configurator database. This approach may cause some problems and, above all, requires the implementation of special mechanisms or procedures to assure coherence between the different databases. Most likely such mechanisms or procedures have to be performed manually, as it would be difficult and expensive to develop a specific interface software and logical structures that have not been conceived in an integrated fashion.

The second possible solution consists in integrating the PDM system with the product configurator. In this way, the configurator technical model and PDM product model partially coincide, avoiding duplicated activities and databases. PDM–configurator integration may allow product design improvements to be promptly fed to manufacturing and sales. In fact, this integration gives the technical staff the option of transferring, in a very simple and fast way, their knowledge to the configuration system.

The advantages of integration are potentially many, but there are still numerous technical problems to be solved before meeting the goal of complete integration. At present, there are some industrial research projects that are trying to integrate PDM and configurators. Some PDM suppliers are including an appropriate configuration module in their systems, or are expanding the configuration capacity of the configuration section within the PDM.

8.3 Configuration systems and Customer Relationship Management

The configurator, as mentioned before, plays a key role in supporting interaction with the customer when customized products are offered. However, the goal of managing the relation with the customer in an integrated way is pursued by the so-called Customer Relationship Management (CRM) systems. Therefore, the question is, how do the functions expected from a configurator relate to the functions expected from a Customer Relationship Management system? Since there is no clear-cut distinction between functionalities of CRM and configuration

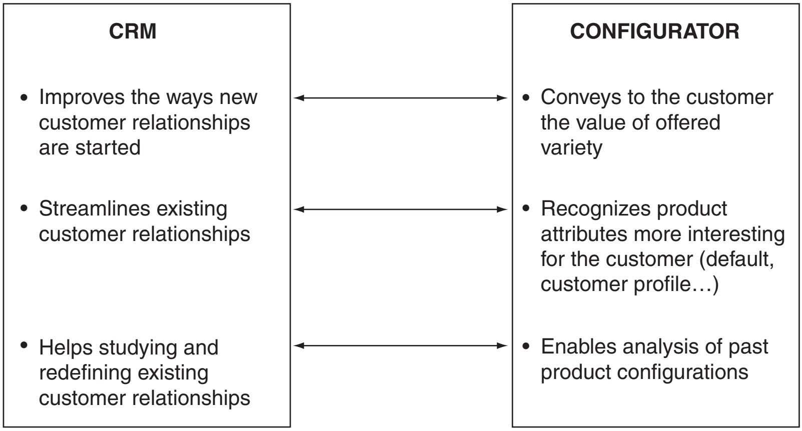


Figure 8.3: Configurator and CRM

systems, the present section aims at identifying points of overlap, of divergence and areas of possible integration between the two systems (see Figure 8.3). This section will briefly explore CRM main functionalities, particularly referring to those contexts with high variety and customization.

Essentially, CRM is a managerial approach focused on the relationship between the company and the customer. The basic idea is that through managing this relationship, which changes in time and is specific for each customer, the company will be able to improve its profits. CRM can obtain such a goal by means of three strategies:

- improving the way new customer relationships are started;
- streamlining existing customer relationships;
- studying and redefining existing customer relationships.

When custom products are offered, the interaction with the customer has to deal with product-related issues as well. After all, customization is intrinsically a relationally intensive activity. Consequently, when product customization is offered, it is unthinkable for a CRM system not to include some of the typical features of a sales configurator (see case 2.1). As shown in Figure 8.3 configurators support each of the three above-mentioned CRM strategies, as will be detailed in the following sections.

8.3.1 Improving the ways new customer relationships are started

According to CRM, the relationship with the customer must be carefully managed from its very beginning. In fact, choosing the appropriate customers and giving them a positive impression from the start, allow the company to save time and avoid the risk of losing potential customers.

To contact potential customers and persuade them to buy is a crucial and difficult task, especially when customized products are offered. Under these circumstances, in fact, it is more difficult to understand what the customer actually needs and to communicate, in a convincing way, what the company is offering (see Chapter 5). Customized products require intensive communication with the customer, which implies longer times and increased efforts in order to establish a customer–supplier relationship.

When a potential customer contacts a company for the first time, he does not have a clear idea of how the product offer may fit his needs. In other words, he needs to learn what product families, variants and options are available. At the same time, he cannot invest too much time and effort in this learning process, as he needs to quickly spot the right product in the ocean of potential alternatives. The sales dialogue of a configurator, indeed, specifically serves the purpose of helping customers to browse through the company's product offer. As observed in Chapter 5, configurators describe the product, structure the interaction with the customer, without overwhelming him or her with options and make the customer appreciate the value of available alternatives. Moreover, the configurators, being highly interactive, foster the customer's learning process.

When the salesman, on the other hand, contacts a customer for the first time, he does not have any precise idea of his or her needs. Contact after contact, he collects more and more information about the customer's needs and preferences. But collecting this information on simple sheets of papers, double-entry tables, pre-printed forms, etc., exposes the company to the risk of collecting incorrect information, losing useful information or forgetting to get the necessary information. This is not the best way to start a relationship. The sales dialogue of a configurator, instead, offers a way out of all these problems. The information collected is correct, as questions should be made unambiguously, and replies should comply with appropriate constraints. Useful information, such as intermediate configurations that did not fully meet customer needs, can be stored and retrieved in case the customer changes his previous choices. Finally, all the necessary information should be collected, as the configurator does not release any configuration until all the required attributes have been specified.

That is why the configurator may support CRM in the acquisition of new customers. Its utility, however, is limited to the aspect 'product', because the configurator does not take into account other elements, such as how far the customer is from placing an order (indifferent, awaiting, motivated, ready to decide).

8.3.2 Streamlining existing customer relationships

Faithful customers are an asset for a company. They place orders absorbing minimal commercial resources, because they are familiar with the company and its products. In addition, they tend to place orders even when the economy is bad, thus providing an ensured income. For these reasons, faithful customers constitute a stock of assets, which the company has a strong incentive to expand and maintain. CRM, accordingly, aims at making life easier to its faithful customers, thus discouraging them from switching to a competitor.

The mechanisms through which CRM streamlines the relation with customers are multiple. For example, when a customer calls to know his order status, the salesman should have updated feedback from factory control, to understand whether manufacturing is on time or not. Were the salesman to notice a delay, he should be able to track down the reason for the delay. He may need an accounting profile of the customer, to understand whether his order has been frozen because of bad credit. Alternatively, he may need to find out who got the order and promised that delivery date. Finally, he may have to check whether the delay was caused by the backlog of paper going to and fro between the customer's and the supplier's technical offices.

Customized products increase the amount of information needed when the company interacts with its customers. More precisely, customization poses to CRM stringent product information management requirements. Prerequisite to the management of information on customized products is the accurate storage of all product specifications concerning past and current product configurations offered to each specific customer. Needless to say, a product configurator provides this capability.

A possible use of past product configurations is in the after-sales service. When a customer needs a spare part for a customized product, the key issue is to identify the right spare part compatible with the specific product configuration in question. For example, if a bathtub manufacturer field service receives a call to replace the pump of a custom showertub five years old, it needs to understand what kind of pump was installed in that product. This may be problematic if the invoice does not indicate bathtub components. If a product configurator was used when selling the tub, by recalling the product code (indicated on the invoice) it would be easy to trace back to the specific pump installed. In case of late or erroneous replacement of the pump, the customer will likely write off the company. As everyday life teaches, it takes a lot to build a relationship and a mere nothing to wipe it out.

Past product configuration data can be also used to speed up configuration activities of faithful customers. Most often, in fact, every customer tends to have his own 'standards', i.e. his configurations tend to share some common requirements. In this case, he expects the company to remember his usual requirements, making life easier for him when placing an order. Product configurators may favour the creation of a customer profile and in this way, the company is able to offer the customer, as defaults, his favoured or typical options. For example, let us consider a manufacturer of gas pressure regulators. If a customer's installations are located in Germany, where the DVGW standard is used, it is possible to create a default that, given the name of the customer, offers him only those solutions complying with DVGW standard. In this way, by automatically screening for those options that apply to his context, he is not required to specify product standard each time.

In conclusion, product configurators support CRM because it systematically remembers and precisely retrieves information on product configurations. Whenever a configurator is missing, there will be a serious risk of mismanaging such information, with negative implications on the customer–supplier interaction.

8.3.3 Studying and redefining existing customer relationships

CRM does not only help in acquiring new customers and streamlining the relationship with active customers. It also helps marketing to set the company's product, price, promotion and placement strategies by providing detailed and structured data on what customers have been asking for, buying, complaining about, etc. In short, it supports marketing planning by means of the most accurate data on customer relationships. How marketing actually uses such data is constrained only by imagination: to screen market trends, to track and improve customer relationships, to assess changes in customers' logistical requirements, etc. For example, a boiler company groups its customers according to their profitability, and based on this value the company offers a series of added services, such as training courses, smoke analysis devices, financial aid to purchase vans, etc. The more profitable a customer, the wider the range of services he can obtain from the company.

Also in this case, product customization adds to the bulk of information CRM has to manage and it increases the complexity of possible analyses as well. To give an idea of possible difficulties, just think of a segment of customers that stopped making orders last year. Let us say they are unsatisfied with the product. What is wrong with the product? What attribute? Or what set of attributes? Etc. Answering these questions implies having an accurate description of those features of products sold in the past to each customer. Product configurators make this information easily available.

A fundamental use of data on customer relations is to calculate how profitable each relationship is. Calculating profitability per relation, when products are customized, cannot be done without precise data on past product configurations. For example, sales data on past product configurations allow the computation of accurate product variable costs, as well as other product costs, e.g. the cost of customer-specific product certifications. Calculating profitability per relationship requires also other non-product related data, outlining the cost of the customer–supplier relationship. Not all relationships, in fact, have the same cost. For example, some customers may take a long time making up their minds about exact product specifications, so that they absorb a lot of the supplier's sales resources. On the other hand, some customers may know the product as well as their needs precisely and, consequently, their decision-making process absorbs very little of the supplier's sales resources. Supported by configuration data, therefore, CRM can help the company to screen customers, supplying information on what relationships have to be fostered or trimmed.

Besides supporting the management of 'customer relationships', CRM systems often include business intelligence tools. Such tools are intended to analyse raw data on customer relationships providing synthetic figures needed by high-level managers. For example, they may supply information on sales by customer segment, by region, etc. When custom products are sold, data on past product configurations are very important in performing such analyses. For instance, by analysing the requests for 'special orders', it may be possible to understand what additional features should be incorporated in the sales dialogue. Alternatively, it may be useful to assess what features are seldom required by customers, so that it may be appropriate to drop them from the sales dialogue. When custom products are offered, therefore, to get the most from the business intelligence tools included within CRM systems, accurate sales data on past product configurations are needed.

9
Selecting a Product Configurator

A company that decides to offer customization through configurable products must face the problem of selecting software capable of supporting this strategy. The market offers a considerable number of software packages, so it may be difficult to understand whether they represent a suitable solution for the company. If so, then the next question is to understand which software is the best. To evaluate alternative software packages, it is necessary to compare the company's requirements with the technical features of available configurators. The present chapter:

- sketches some of the key questions in determining the requirements placed upon a product configurator;
- identifies the fundamental groups of technical features to describe a product configurator;
- illustrates each technical feature of a product configurator.

9.1 The configurator: defining expected benefits and requirements

Once a company has decided to offer customized products with an approach based on product configuration, it has to choose the information system that supports such an approach. This is not a straightforward task. The company has to understand what is going to be configured: fabrication, assembly or distribution (see case 1.4). Furthermore, different product families may require different types of configuration. Finally, to aggravate things, the configuration logic may be applied only partially (see case 2.4) to some product families.

Economic considerations are needed to complement the definition of the 'ideal' requirements of the product configurator. The convenience depends, on the one hand, on the advantages in terms of better efficiency and services that customers may obtain and, on the other hand, on the costs related to the acquisition, set-up and maintenance of the configurator. The advantages deriving from a configurator depend on:

- the number of configurations to be made in the unit of time;
- the length of the activities related to the single configuration;
- the costs and risks born from erroneous configurations;
- the customer requirements in terms of speed, precision and support for the configurational choice.

The costs of a configurator are higher if:

- the products are complex;
- the product families are numerous;
- the products are deeply customized;
- the product parts to be customized are numerous.

The company business environment and the way it decides to shape its configuration process impose the overall requirements over the configuration system. These requirements have to be considered when evaluating alternative commercial packages. If the company is not clearly aware of these requirements, it runs the risk of selecting, and probably buying, a system that is insufficient or oversized for its own needs. In both cases, the wrong choice will imply, at least, excessive costs because the system must be customized or, in the case of an inadequate system, because it does not comply with the real needs.

The problem of choosing a configurator is not to find the best one, but to identify the one that best suits the needs of the specific company. If the configurator does not fulfil the requirements of the company, the advantage will be reduced or even cancelled. Clear ideas about objectives and requirements allow the company to compare the configurators the market offers and finally make the most appropriate choice.

The evaluation is very complex, because there are many features to consider. In order to analyse the various configurators, it is useful to follow a logical scheme that takes into account the different characteristics offered. We suggest the following list, that classifies the parameters into five fundamental headings:

- basic functions;
- specific functions;
- configurator technology;
- modelling approaches;
- interaction with the user.

In the general evaluation of the software, each company will give different importance to the different parameters of the list, considering not only the benefits of the system, but also the agreement with its own objectives. The next paragraphs present a detailed description of all the parameters. Obviously, the combination of all these elements may lead to very different situations, which require different software supports for product configuration.

9.2 Basic functions

The functions of a configurator can be divided into two groups, according to their presence and importance in modern configurators. The first group includes basic functions: if a configurator does not have such features, it cannot be considered a modern configurator. The second group includes special functions, i.e. those functions that are useful in specific situations, but that are not always needed in a configurator.

Table 9.1: Basic functions of modern configurators

Product information supply	The system generates all the product information the customer needs, with special reference to choices available to customers
Generation and retrieval of commercial configuration	The system collects customer choices, searches for already existing commercial configurations or generates new configurations complying with customer choices and product constraints set by the company
Generation of technical configuration	The system creates the documents necessary to manufacture the product (bills of materials, production cycles, etc.)
Commercial modelling	The system supports the creation and modification of a sales dialogue
Technical modelling	The system supports the creation and modification of a technical model

First of all, it is necessary to consider the basic functions, which represent the most immediate cause for choosing a certain configurator. Table 9.1 lists and describes these basic functions.

The basic functions can be considered in relation to the two elements that characterize their architecture: configuration engines and modellers. The first three functions described in Table 9.1 are performed by the configuration engines; some of them concern the commercial area, others the technical area. The last two functions are carried out by modellers. There follows a detailed description of the functions shown in Table 9.1.

Product information provision The commercial configuration engine must be able to provide all the information about product characteristics used in the sales dialogue. In particular, it must give information about the general product characteristics and about customer choices, automatically controlling the feasibility of each proposal.

Generation and retrieval of commercial configuration The commercial configuration engine must guide the user in product selection, helping the customer to identify those product characteristics that best fulfil his needs. In doing so, it must gather all the information necessary to identify a certain product configuration from the commercial point of view. Afterwards, the commercial configuration engine must help the company staff to search among already existing configurations that can satisfy customer requirements. If these requirements do not correspond to any of the previous configurations, the engine must support the generation of new configurations, that respect the imposed company limitation, and must store them for further uses.

Generation of technical configuration Once the variant that satisfies customer needs has been identified, the technical configuration engine must be able to provide the documentation necessary to produce such a variant, for example, product cycle and bill of materials. According to the context, other documents may be requested, such as product lay-out, drawings, manuals, etc.

Commercial modelling Configurators allow definition of sales dialogue structure: definition of questions and association of possible answers to the constraints they have to satisfy. Very often, modellers also give the possibility of connecting phrases, images, drawings or clips to the questions, in order to help the interaction between customers and sales people.

Technical modelling The elaboration of the technical documentation requires the existence of a technical model. This model includes product characteristics, product components, the different values such characteristics may assume and the constraints among those characteristics. The technical model is essential for a correct use of the configuration system. Based on this model, the technical configuration engine performs automatic computations. The creation and updating of this model may be very costly if they are not adequately supported by a modeller.

9.3 Special functions

In the previous paragraphs we described the most important functions, which are present in the majority of the configuration software on the market. There may be other functions, which can be very useful,

Table 9.2: Special functions

Generation of the offer	The system generates offers including product characteristics, additional services, price and other commercial conditions
Production process configuration	The system can configure production cycles
Coding	The system automatically generates product codes according to multiple criteria
Impossible configuration management	The system may allow the user to input attribute values that do not satisfy all constraints
Management of indifferent answers	The system does not require the user to make a choice for product attributes he does not care for
Default values	The system proposes default values for classes of customers or it drives customers towards 'standard' options
Document customization	The system allows the user to define the format of the documents it generates
Multiple languages and currencies	The system can describe products and prices in multiple languages and currencies
Integration with MPCS	The system supplies feasible delivery dates based on information from the MPCS
Mathematical functions	The system can represent complex constraints between different product attributes

according to the company context. These functions are likely to differentiate product configurators. They are summarized in Table 9.2.

Offer generation After selecting the commercial configuration that best suits the specific customer needs, the commercial configuration engine must be able to provide precise and fast information about the price of the requested variant. Moreover, it must be able to give the customer the commercial description of the variant selected, as well as the usual commercial information, for example, delivery terms, that completes the offer. The offer may be complemented by the description of additional services bundled with the product that the customer is requiring.

Production process configuration The system allows the possibility of configuring, along with the products, the different production and assembly processes. This function is normally requested when manufacturing activities vary according to the type of configured product. In these cases, production and assembly processes need specific information to elaborate the requested variant.

Coding The system gives the possibility of handling the codes already existing in the ERP database and creating new ones applying the coding rules required by the company. Although the coding function is nowadays present in every configurator, there may be some limitations in coding rules. Such restrictions may become a difficulty when introducing a configurator.

Impossible configuration management Normally, during the sales dialogue the different configurators propose only the options that are compatible with the values already established. Yet, this is not the only available way to manage the sales dialogue. In some cases, the system might allow the customer to choose any option, and then indicates the presence of possible incongruence, compelling the user to modify his preferences. In this way the user is given an explanation of the impossibility of fulfilling his original requirement, or the salesman has an explanation prompt for the customer. Furthermore, the system may complete the configuration even in the case of unacceptable options. In this case some constraints are ignored, leaving the decision to manufacture the requested product or not in the hands of the designer. This functionality is useful when a customer or an order is particularly important for the company.

Indifferent answer management The customer may be indifferent to some of the product attributes, therefore the system must be able to manage this situation. The most frequent solution is to assign default values to these characteristics; values that must be coherent with the existing constraints. Another alternative may be the possibility of presenting all the potential solutions to the customer, leaving up to him the choice of the criterion selected to determine the best configuration. In other words, we may have a product described by six attributes A, B, … F. A customer may be interested in specifying only A, B and C and not D, E and F. However, he may be interested in having the system choose for him D, E and F in order to have, for example, the product as soon as possible or, alternatively, at the lowest possible price.

Default values Default values are not only used to manage indifferent answers in the sales dialogue, but also to communicate recommended answers, leaving the customer free to accept or reject the defaults later. In this way, it is possible to influence the customer, compelling him to choose one solution rather than another one (perhaps more customized). Default values generally are divided into two types: static and dynamic. In the first type, the value is assigned during the modelling dialogue and cannot be modified. A dynamic value, instead, is redefined by the system according to the answers given during the configuration dialogue. In synthesis, the possibility of defining default values is very important, because it helps to reduce the risk of customers asking for things they do not really need, whilst they are left free to choose whatever they deem important. By doing so, useless product proliferation is reduced, with positive operational benefits.

Document customization It is possible to include the company logo, or the name of the retailer or salesman who is carrying out the commercial transaction, in the budgets, contracts and reports. This function does not aim at a particular advantage, because it only influences the aesthetic aspect of the documents created by the system. However, it contributes to confer an image of quality and seriousness to the company.

Multiple languages and currencies This is an indispensable function for companies that operate at an international level. With the growing trend of globalization, almost all the configurators on the market give the possibility of offering prices and descriptions in multiple currencies and languages. Yet, it is always better to check the real capabilities of the system.

Integration with Manufacturing Planning and Control System (MPCS) This function integrates the Production Planning and Control system with the configurator. The degree and ways of this integration depend on the type of configurator chosen (integrated or stand-alone). If they are tightly integrated, during the configuration process, the configurator might have information about possible delivery times from the MPCS. On the other hand, if they are not integrated, it is important to check that the basic data exchange (i.e. BOM and production cycle) operates correctly.

Mathematical functions They normally include the fundamental mathematical operations and the Boolean, trigonometric, logarithmic and exponential functions. These functions may be useful to express, in mathematical terms, the constraints between different components. Due to advanced functions, it is possible to introduce complex constraints that, sometimes, are needed to get a product configuration.

9.4 Configurator technology

A third group of technical features to consider, when selecting a configurator, is the system technology. Table 9.3 shows the most important aspects that may differentiate the various configurators.

Table 9.3: Configurator technology

Modular architecture	The system is structured in independent modules
Synchronous connection	The system exchanges data in real-time with other connected software applications
Type of implementation	The system may use different formats and methods to communicate information between server and client
Client–server model	The system may have a different balance between client-side and server-side
Integration with the ERP	The system can have different degrees of compatibility and data exchange capability with the company ERP
Availability of source code to the user	The system source code can be accessed by the user for in-house development or modification
Adaptability	The system is compatible with different hardware, operative systems and databases

Modular architecture A modular architecture, unlike an integral one, allows attaching special functions (see case 9.3) by adding specific modules to the basic configuration software package. In this way, it is possible to buy or implement only the functions needed, and later expand them based on needs and availability of resources.

Synchronous connection The connection between the configuration system and other applications may be synchronous or asynchronous. In the first case, data are updated and shared in real time, avoiding possible incongruence between different applications. In the second case, the information exchange is not continuous but carried out through updating batches. This type of connection is cheaper, but the delivery date cannot be directly controlled during the order acquisition process (although this problem may be satisfactorily solved using proxies). It is advisable to use the asynchronous connection when the company has a mobile sales force, and the data necessary to create a budget or generate an order do not have to be updated in real time.

Type of implementation According to the choices made in terms of final user and sales channels, the types of implementation adopted may be: *mobile*, *Internet HTML* or *Internet client–server*. In a *mobile* implementation, the graphic interface, the configuration engine and the different models are set in the portable PC, and the salesman can configure the product without being connected to the net. Yet, periodically, he has to connect his PC to the company network to update the information about orders and probably about configuration models. In an *Internet HTML* implementation the graphic interface for the user is like a web and consists of a form that any kind of browser can visualize. In order to configure a product, the user has to get access to the server where the configuration engine is set up and, since the system works page to page, he has to complete a page of requests before getting any feedback from the system. Finally, in an *Internet client–server* implementation, the graphic interface for the user is dynamic since it is dynamically loaded by the user's browser at the beginning of the configuration process: unlike in the preceding case, to configure a product the user has to connect to the server, but the system provides feedback after each click.

Client–server model The three-tier architecture of a client–server system considers the subdivision of a software application at three levels: *user service* (that manages the images, controls, graphics and messages shown on screen), *business service* (that implements the program application logic, i.e. calculations, data verification, operations) and *data service* (that manages the access to data and controls their insertion, maintenance and research). Remaining within the client–server architecture, it is possible to have a different balance between the data *on the server* side and those *on the client* side. For the client, the updating requirements may be different from those of the server.

Integration with the ERP This is the possibility of integrating a certain configurator with the computing system used by the company (see case 8.1). This aspect must be considered not only in the case of a stand-alone configurator, but also if the configurator is part of a wider ERP system. The possibility and the necessity of exchanging information with other system modules must be accurately examined, because integration costs could be very high or certain functions might not be performed.

Availability of source code to the user This is the possibility of obtaining the source codes of the configurator from the software vendor. This feature is interesting for companies that have an IT staff capable of modifying the configurator, and are interested in directly adapting the software to their needs. In these cases, the program language used to elaborate the configurator must also be considered. If it is a widespread language, it will be easier to find programmers who will program and maintain the configurator.

Adaptability This is the possibility of adapting the configurator to different kinds of hardware (AS400, Sun, etc.), software (Windows, Unix, etc.) and databases (Oracle, DB2, etc.) This aspect is very important, because it allows the use of existing infrastructures, reducing times and costs to make the system operative.

9.5 Modelling approaches

The next group of technical features to be considered, when selecting a configuration software system, involves the available approaches for technical modelling. Such features are displayed in Table 9.4.

Single or multi-level configuration This feature refers to the configuration of complex systems made up of a set of subordinated components, where each component is in itself configurable. In the case of single

Table 9.4: Modelling approaches

Single or multi-level configuration	The system allows the creation of a configuration profile of a complex product, and then separately configures each constituent component
Single or two-step configuration	The system permits product data inputting only during the commercial configuration process or during the technical configuration process as well
Parametric component configuration	The system can configure components whose parameters vary within a continuous interval
Semi-finished component configuration	The system can automatically generate new semi-finished components
Method for computing product cost	The system allows defining costs only for elements at the first level in the BOM, or for all product components
Method for computing product price	The system allows pricing the product by applying a mark-up on product cost or by adding up prices related to options on the sales dialogue
Approach to model creation	The system may embed different mechanisms for product model creation and updating: rule-based, object-oriented, etc.
Support of multiple sales dialogues	The system handles multiple sales dialogues oriented to different customer targets
Special or general-purpose configuration	The system is specifically conceived for a certain kind of product or can be applied to different kinds of products
Product or user-oriented configuration	The system's sales dialogue should mirror the technical product model or it can be defined independently from the technical model

configuration (or configuration at a single level) the configurator allows configuring only the main product, that is to say, the product sold to the client, without personalizing the components that constitute the final product, since such components are selected according to the values assigned to the main product features. In the case of multi-level configuration, it is possible to configure the single components (or subordinated products) independently, after obtaining a configuration profile for each of them, according to the macro characteristics of the system. A multi-level configuration offers considerable flexibility in the case of products constituted by various components that, one by one, have to be configured. For example, a professional kitchen involves a system-level configuration, where you have to specify layout, kitchen module depth, power inputs (electrical supply, gas type and pressure), country regulations, etc. The kitchen components, however, have themselves to be configured. For example, the oven may have a steam-boiler or just the heating resistors, the control panel may be digital or analogic, etc. Of course, most of the system-level specification will affect the oven, as well as all the other kitchen components.

Single or two-step configuration This is the possibility of structuring the configuration process in two steps. In the sales dialogue all the information that describes the product, from the commercial point of view, is gathered. In the second step, the technical office personnel complete the product description from the technical point of view. This further definition is necessary in all those cases where there is a 'distance' between product functions and product manufacturing characteristics. In fact, in such cases, the sales dialogue is not enough to gather all the necessary information to generate the real bill of materials, therefore a technical specialist is needed, since he is able to make 'expert' choices, within certain degrees of freedom, and elaborate the complete product definition.

Parametric component configuration This is the possibility of managing components whose parameters change continuously. Typical parameters of this kind are of a dimensional nature (length, width, height, diameter, etc.). For example, if we have to configure a round table, the radius may have different values. In this case the component 'top surface' is linked to the parametric variable 'radius'. Obviously, it is necessary to define the extreme values for each parameter, which correspond to each component.

Semi-finished component configuration This is the possibility of generating codes for new semi-finished products. This possibility is useful when a new semi-finished product is produced: usually, it happens when the semi-finished product is a sub-set obtained by means of a high number of components or when there are dimensional features that can change based on customer request. The possibility of automatically creating a code for the new semi-finished product allows it to be identified in the bill of materials and is useful for a more detailed product schedule and a more accurate production control. When an element of the generic bill is associated with a codifying rule that, each time, creates the real element and inserts it in the configuration, such an element is called self-codifying (see case 6.5).

Method for computing product cost The total cost of a configured product is obtained by adding up the costs of the single elements present in the BOM including the production cycles costs. To calculate this total sum, it is necessary to know the cost of each article and to adopt a criterion to estimate production cycles (direct cost, full cost, etc.). Alternatively, it is possible to associate a cost to each element at the first level of the BOM and add them up. But, in this case, it is necessary to determine a cost for all the possible elements at the first level.

Method for computing product price There are several criteria implemented in different configurators to determine finished product prices. In the *article price list* criterion, a price is assigned to each finished product. In the second criterion, the price list is *based on the elements of the configuration dialogu*e and the finished product price is determined adding up the values assigned to each answer given in the course of the dialogue. In the third criterion, a price is assigned to the *first level elements of the BOM* and the addition of all these prices constitutes the finished product price. In the fourth criterion, based on the BOM costs, the price is obtained applying a *mark-up to the costs* associated to the different elements of the BOM and to the various production cycles.

Approach to model creation The way in which the technical model is created is one of the most important features of a configurator, because an easy updating of the technical model and the capability to rapidly expand it to new products depend on this characteristic. An easy updating and a rapid expansion are attributes that everyone longs for, but at the same time these attributes present a series of problems, particularly for the intrinsic difficulty in determining under which conditions a model is easily updated. Two interesting parameters must be considered: the initial training hours of the system operator and the average time necessary for each action performed on the model. Independently from the type of approach adopted by the configurator, it is necessary to analyse some aspects that characterize the system implementation and updating: (a) how to describe the relationships between components, (b) how to describe the constraints and incompatibilities of the different solutions, (c) 'bottom-up' or 'top-down' solution, i.e. the components are added one at a time or starting from an abstract description; such description is afterwards put into context and upgraded in successive phases.

The simplest approach, which was the first one applied in the creation of a configurator, is called *features and options*. In this approach all the components, used for a certain product family, are classified according to different groups of technological products (in the case of a PC we may have: memories, hard disks, video cards, etc.). These classes (or features) are organized according to a hierarchical structure: each article, independently of the class it belongs to, may require the insertion of mandatory or optional sub-classes. When a certain class is inserted in a configuration process, all the articles belonging to it become possible choices. The configuration process consists in choosing, for each class, the most appropriate component, remembering that the features indicated as options can be omitted. The process finishes when all the classes have been examined. In this approach, the various constraints are not declared, but they are implicit in the structure of each class, and this implies some difficulties in model updating.

In the *rule-based* approach, a technician devoted to creating the technical model defines a series of rules of the type 'if … then'. Later on, during the compilation phase an inference engine reads the rules, applies them according an appropriate strategy and involves the user when the process requires further choices. The user interface typically consists of a sequence of questions, whose order is established by the inference engine. Rule-based systems are able to solve more complex situations, compared to 'features and options' systems, for example, situations that involve dimensional calculations and the management of policode codes. Moreover, they allow for explaining the process, i.e. they tell the user why a particular question is asked. However, since each single option may have its own rule, the total number of constraints could be over a thousand. If the number of constraints increases, the updating process becomes more complex, because the consistency between the rules is guaranteed by the inference engine, not by the code sequences of the program. To make things worse, the difference between rules and data is not very clear, since article codes and options are somehow hidden, rather than stored in the database according to a rational logic.

The third possible approach is known as *object-oriented*. Components, standard products, configurations and any other type of elements are represented by means of objects that have their own identity. Each object is characterized by a certain number of attributes, which permit one to identify unmistakably the single objects. An object may have constraints that specify the conditions for its suitability. During the configuration process, various objects may be considered valid and finally a particular configuration of the requested product is adopted. These types of configurators are able to deal with most situations, but an expert programmer is required to codify the logic and the user interface. Very frequently, product engineers are not qualified enough to keep pace with the system. The implementation becomes very complex and costly, which is why this approach is nowadays mainly used at research and prototype levels, and is not widespread.

The fourth approach is called *table-based*. This approach differs, as far as complexity, from *rule-based* and *object-oriented* approaches. It was adopted in the first front-office configurators, setting rules by means of tables. It is very similar, in its pros and cons, to the *features and options* approach. On the one hand, it is easier to implement and operate, even for non-technical personnel. On the other hand, it is limited in those applications where continuous parameters are needed. If the tables have a predefined structure, updating techniques are ergonomic and well designed. Unfortunately, systems exclusively based on this approach have limited applications, in the absence of appropriate customization.

The fifth approach is known as *constraint-based*. This approach started in the early 1990s, with the purpose of correcting the defects of the rule-based technique. The basic idea is that, very frequently, it is easier to control a non-valid solution, rather than a valid one. These systems adopt an inference engine based on abduction logic rather than on deductive logic (as in the case of rule-based systems). According to this reasoning, if A implies B, and direct observation says that B is true, then we may conclude that A may have caused B. Configuration and engineering design are more similar to abduction than to deduction. When a designer wishes to carry out a function F for a certain product and his experience suggests adopting the component C, because it is the most suitable one, by applying abduction he chooses component C. If in the successive phases of the project some incongruence appears, the designer will substitute component C for another component that also carries out F.

The market offers a great variety of *hybrid systems*, i.e. solutions that combine different approaches. These systems try to improve, through different solutions, the rule-based philosophy.

Support of multiple sales dialogues This is the possibility of having contemporaneously different sales dialogues, interacting in the best possible way with different types of users. For example, the same product may require a more dynamic dialogue, full of images and comments, if the product is to be sold in a self-service fashion. On the contrary, the dialogue may be more important if the salesman has a good knowledge of the product and enters data in person, e.g. while policode with the customer on the phone. Offering more than one type of dialogue fundamentally helps to serve different users and sales channels.

Special or general-purpose configuration Special-purpose configurators were the first ones on the market. They were created as an answer to specific product requirements, and consequently they were often developed 'ad hoc'. It is difficult to adapt them to different contexts. That is why they are used where technology and design rules are consolidated and relatively stable. Nowadays, general-purpose configurators are the most widespread. In fact, because many products appear, develop and disappear rapidly, it is more and more important to model different product families, minimizing the efforts and costs involved.

Product or user-oriented configuration In *product-oriented* configurators the technical model guides the gathering of information, i.e. the sales dialogue mirrors the generic bill of materials. On the contrary, *user-oriented* configurators are designed to accomplish the interactive dialogue with the user independently from the structure of the bill of materials. A product-oriented configurator is more suitable than a user-oriented one, as long as the customer has a good knowledge of the product.

9.6 Interaction with the user

Finally, the last group of technical features to be considered, when selecting a configurator, refers to the different ways in which the user interacts with the system. Apart from being easy to use and to update, the system may also present other specific aspects, as shown in Table 9.5.

Ease of updating New variants are continuously introduced due to customer demands. Consequently, updating the system must be an easy task. Information systems must offer tools that simplify technical modelling processes, even in the case of complex products. Therefore, it is

Table 9.5: Interaction with the user

User-friendliness	The system is perceived as easy to use by its prospective users
Ease of updating	The system includes tools and interfaces to make updating of commercial and technical models easy
Sales dialogue interactivity	The system allows modification of an answer at any time during configuration dialogue
Multi-session product configuration	The system allows starting from previous or partial configurations previously saved
User interface specialization	The system can present different user interfaces, depending on the type of user (salesman, agent, distributor, non-expert customer, etc.)
Graphic interface versatility	The system user interface can be easily adapted to the user requirements
Output format versatility	The system outputs can be easily adapted to the user requirements
Product configuration comparison	The system allows comparison of different configurations to identify similarities and differences
Product configuration retrieval versatility	The system permits the retrieval of past configuration based on alternative criteria (customer, common parts, geographic areas or price)

extremely important to use a modeller that simplifies the way in which technical models are created and modified. A modeller, as mentioned in previous chapters, overcomes an important defect of special-purpose software, that is to say, its limited flexibility. In fact, the impossibility of separating model product and configuration process, forces the manual rewriting of part of the system software code whenever it is necessary to modify an existing product. On the contrary, modellers are easily handled by the company staff and require elementary programming knowledge: in some cases, the module presents a graphic interface, and so the modification of a technical model does not require any programming activity.

Sales dialogue interactivity This feature is the possibility of reconsidering, during the configuration process, some of the decisions taken, without re-entering all the choices. This feature makes the sales dialogue particularly resourceful and flexible, thus making the configurator more profitable.

Multi-session product configuration In the case of very complex products (for example, an airplane cabin) it may be useful to use inputs for the configuration process, some partial layouts, obtained in previous configurations. Yet, this procedure is not widespread and is typically applied to very particular types of software.

User interface specialization This is the possibility of supporting different users, such as internal technical personnel, salespeople, agents or customers belonging to different segments. For prospective customers, it will be necessary to limit the available information. Moreover, different kinds of dialogue may be necessary if a company has customers with different degrees of product knowledge. This is especially true when a company uses a wide range of sales channels (call centres, web, large distribution, etc.).

Graphic interface versatility The type of graphic interface used is very important, especially when the configurator user is external to the company. There are many types of interface, but the important most are the following.

In the case of *sales dialogue interface*, the questions are asked one at a time and depend on the preceding answers. This type of interface is particularly useful in the case of telemarketing, where the user speaks on the phone with the customer.

The *catalogue interface* solution allows customers to specify the components they need. It is suitable for simple products, but not recommended when we want to describe the product usage (for example, not all customers may be able to tell which components are required for a PC used by children with limited Internet connections).

In the *needs analysis interface* the dialogues that analyse the different needs are shown as tables. To select, it is necessary to use drop menus, filling matrix and choice buttons. This solution expresses the concepts in an intuitive way, but may require a complex process for product conceptualization.

The *CAD or schematic drawing interface* allows the user to interact with a graphic view of the problem shown in two or three dimensions. Also, these graphic interfaces are easy to use, but require long development and updating times to interact with the configuration engines.

Output format versatility Once the configuration process is over, results can be presented to the user in different formats. These outputs may include a component list, drawings, images or formulas or formatted in the appropriate way. Being able to easily change or update output format may serve commercial purposes, increasing sales productivity and effectiveness.

Comparison of product configurations This is the possibility of comparing various configured products to identify differences in terms of bill of materials components, and product specifications. This function improves dialogue versatility and gives the customer a more detailed service in terms of product analysis. Moreover, the comparison is useful when the production department has to evaluate the possibility of using a product configured for a specific customer, to satisfy another 'urgent' order received subsequently from a more important customer.

Product configuration retrieval versatility This is the set of criteria for searching the database of past configurations. It may be useful to carry out research focused on (A) a customer, to know everything a certain customer has bought; (B) a date or period, to check how the characteristics of the different orders vary throughout a year; (C) a certain geographical area, to verify how requests vary according to the region; (D) the presence of common parts, to know the popularity of a certain component within the various finished products and to estimate future demands; (E) a certain price segment, to understand what features are typically requested within a certain price segment. If the company does not have a business intelligence package to analyse past product configuration data, versatility in retrieving past product configurations may partially serve the purpose of performing basic product analyses.

10
Implementing a Configuration System

The implementation of a configurator is a costly process that requires technical and managerial considerations. Improper implementation management may lead to delays, excessive costs, and sometimes, to failure. Therefore, it is important to have sound guidelines to plan and execute such a task. This chapter points to three key activities:

- to analyse what are the benefits of the configurator and to contrast them with its costs;
- to plan for configuration implementation;
- to execute implementation activities following the best implementation practices.

10.1 A reference process

Implementing a product configuration system may be done in many different ways. Nevertheless, a logical sequence linking the different activities needed to achieve this objective can be defined (see Figure 10.1). Such a sequence should be respected, because it favours the gradual, harmonic and disciplined development of the project.

The *preliminary analysis* is a rapid, global investigation, with the purpose of understanding whether product customization is a problem to be addressed by the company, or not. This analysis aims at

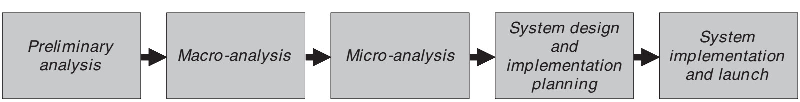


Figure 10.1: Different steps in the implementation process of a configurator

describing the customization strategy of the company: what customization the market is asking for, how customization is offered by the company, and more generally what opportunities and constraints are offered to the company by product customization.

The *macro-analysis* is a comprehensive survey on the different configuration problems in the company being studied. The aims of this analysis are: to define, at a macro level, how configuration processes may be modified with the adoption of a configurator, to quantify potential advantages and to identify which product families must be considered first.

The *micro-analysis* is a detailed examination of the current configuration process and of how it can be redefined with the support of a configurator. The objectives of this analysis are: to assess whether the configurator is technically and economically feasible. The macro-analysis defines system specifications and evaluates its costs and advantages.

In the *implementation decision and planning* step, we must decide whether to continue with the project. The software is selected and its implementation is planned. The aim of this step is to decide whether the product configuration system is to be implemented and, if so, how it is to be implemented.

In the *system implementation and launch* step, the different models (technical, commercial, etc.) are created using the software already selected, for all the configurable product families. Then, such models are tested and upgraded until the configurator starts working at full capacity. The aim is to 'build up' the system and put it to work.

The length and costs of the different steps are highly variable and depend on the particular characteristics of the environment. This analysis process may be more or less accurate and detailed, according to the managerial style adopted by the company. Obviously, if there is too much hesitation during the analysis, the company runs the risk of wasting too many resources and of having a considerable delay with regards to its competitors. Conversely, a superficial and rapid analysis may lead to a project that shows deficiencies in system implementation and launching or, later on, when the system needs to be overhauled.

However, the process may be developed in a different order, for example, the decision to buy the software may be made earlier, with respect to the sequence above-mentioned: in some cases, a company may be certain about the effectiveness of a configurator, because a competitor who uses it has significant advantages.

10.2 Preliminary analysis: do we really need it?

The rapid preliminary analysis is made at a global level to understand whether configuration problems need to be researched on the part of the enterprise. Figure 10.2 shows the aims, activities and deliverables related to this step.

Outlining the customization strategy of the company. Above all, the preliminary analysis outlines the configuration strategy of the company in question. In fact, the company may have a great number of products with a long life cycle, as in the case of pipe fittings, or products with a short life cycle and with a configuration that depends on highly variable customer requirements, as in the case of industrial AC motor drives. It is possible to have very complex products, manufactured in small quantities, with technical characteristics that vary considerably according to customer requirements, as in the case of industrial machinery. A correct description of the customization strategy, therefore, must be the first step, because it helps the company to ask the appropriate questions.

In the preliminary analysis, the first activity consists in gathering information about variety and customization. It is necessary to identify the number of product families, models and variants the company has produced or is likely to produce. These data, together with the information about the average number of annual configurations, give us a picture of the situation under analysis. Apart from this static datum, it is important to record a dynamic datum about variety and customization, because it provides information on the flexibility required in the system.

Understanding whether and to what extent the configuration process represents a problem. The second aim is to understand whether the configuration process is really a problem for the company and if it is a relevant one. The problems in a configuration process may be sporadic faulty events or frequent and systematic mistakes with serious consequences. It is obvious that configuration process improvement becomes more important in the presence of numerous and severe performance problems.

The second activity is a survey on the problems related to the configuration process, identifying their severity and their consequences. At this point a series of parameters is useful, such as the rate of configuration mistakes in customer orders, the rate of mistakes

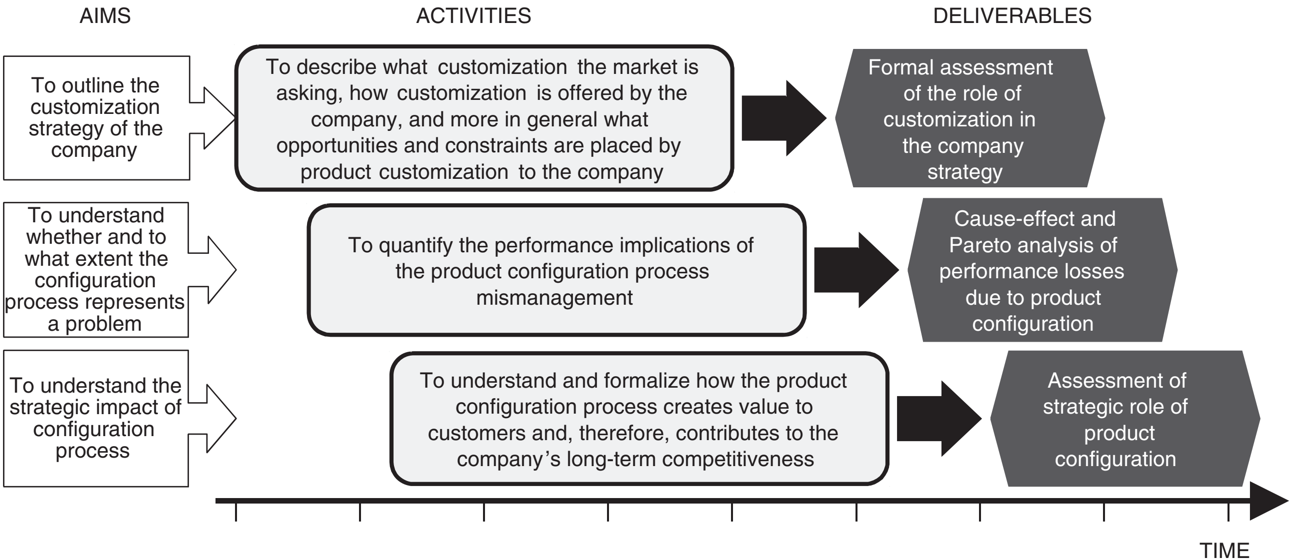


Figure 10.2: Preliminary analysis: aims, activities and deliverables

in bills of materials, the rate of configuration mistakes in production orders, etc. Resources absorbed by product configuration process as well as its lead-time are also recorded during this phase. A possible indicator of resource consumption in the technical office could be, for example, the total time devoted to elaborating bills of materials.

Understanding the strategic impact of the configuration process. The third aim is to understand how the configuration process impacts the company's competitiveness. If the company is heavily squeezed between the contrasting requirements for customization, low product costs and fast product delivery, then the configuration process becomes strategically relevant.

Finally, problems in the configuration process are compared to the critical reasons for succeeding on the market. This is achieved by considering the key success factors separately and analysing how the current configuration process supports or reduces company competitiveness. For example, if the market demands timely offers, but the company configuration process takes a long time to elaborate them, the impact of the configuration process on competitiveness is high and negative. In this case, configuration problems are certainly relevant, from the strategic point of view. If some key success factors on the market are negatively influenced by configuration problems, it is necessary to continue with a deeper analysis.

Preliminary analysis is usually very quick and it requires only a couple of days to be carried out by a person who knows the configuration process. It may be an employee with some knowledge on the subject, or an external expert who takes a first glance at the company.

10.3 Macro-analysis: to what extent do we need it?

The macro-analysis is a comprehensive appraisal of the problems related to the configuration process in the company under study. This analysis differs from the preliminary analysis in two respects: firstly, it is carried out on each product family, rather than at a global level; secondly, it goes beyond the symptoms and opinions described in the preliminary analysis and is based on more reliable information. Figure 10.3 shows the aims, activities and deliverables related to this analysis.

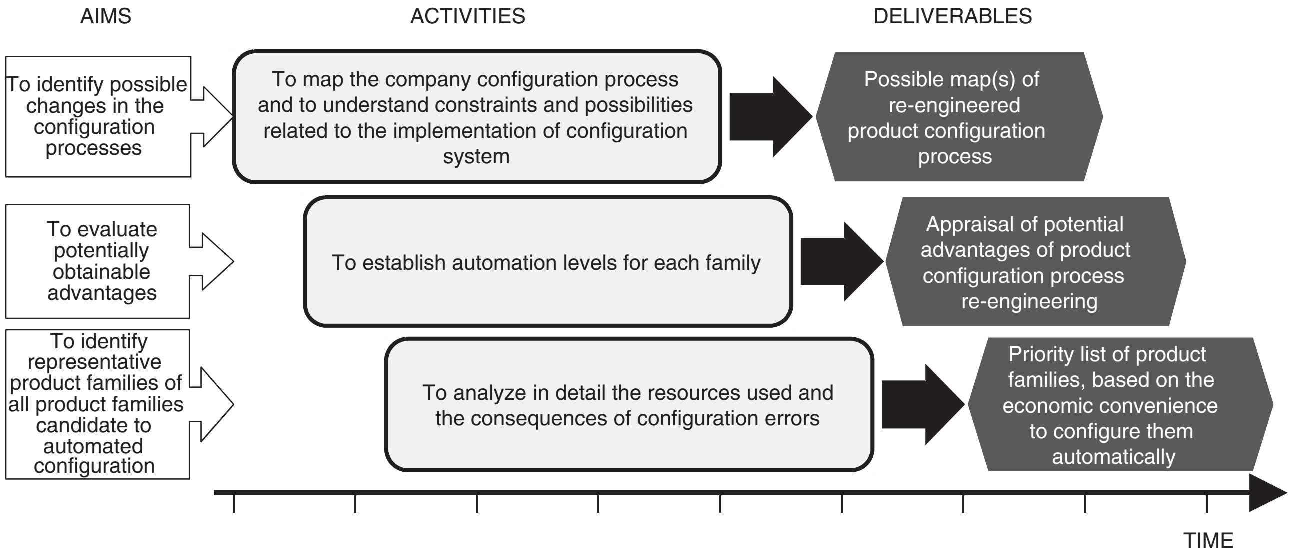


Figure 10.3: Macro-analysis: aims, activities and deliverables

Possible map(s) of re-engineered product configuration process. The first aim is to define, at a macro level, how the configuration process could be modified by using a configurator. Above all, the purpose is to understand whether it is possible and reasonable to differentiate the configuration process, i.e. different configurations for different product customizations. For example, there will be different processes if the product strategy is pure customization, customized fabrication or customized assembly (see case 1.3). Then, it is necessary to understand how the sequence of configuration activities is affected by the introduction of a configurator.

The macro-analysis starts with the detailed mapping of the company configuration process. It is necessary to consider all the activities of the configuration process and to analyse how they are really carried out. This helps one to understand which activities can be modified by introducing a configurator. At the same time, this analysis pinpoints what essential company requirements the configurator will have to comply with. This analysis gives one an idea of how information tools and company processes have to be aligned.

Appraisal of potential advantages derived by product configuration process re-engineering. The second aim is to evaluate the potential advantages of adopting a configurator. Both objective and perceptive advantages have to be considered. Objective advantages may be, for example, lead times and cost reductions. A perceptive advantage may be, for example, improvement of company image on the part of the customer.

The automation of the various activities does not produce the same benefits and does not require the same efforts for all products. Therefore, it is necessary to consider the different configuration activities for each single product family and then estimate which ones may be automated and which ones may not. Automation limits must be identified, even though approximately. It is important to determine if and to what extent the various configurations follow repetitive schemes, and whether and to what extent it is possible to formalize a sales dialogue. It will be useful to interview the people in charge of technical and commercial offices, because they are able to answer those questions.

Identifying representative product families. The third aim is to identify priorities among the different product families. In order to accomplish this purpose, it is necessary to compare costs and advantages of adopting a configurator for each product family. Ideal candidates for automatic configuration are either simple product families involving a great number of configurations per year, or complex families configured according to a repetitive scheme. In contrast, the less appealing candidates for automatic configuration are either simple families with a low number of annual configurations, or extremely complex with very few repetitive elements. The principle is that the benefit of adopting a configurator depends on the number of annual configurations and on the advantage derived by supporting each individual configuration. The implementation cost of a configurator, instead, is related to the complexity of product family description.

Finally, in order to assess the potential benefits derived from the adoption of a configurator, it is necessary to measure, in detail, the resources used by the activities of the current configuration process, especially those activities that can be automated. Moreover, it is necessary to identify the various errors of the current configuration process and to evaluate how many resources could be employed elsewhere, if such errors were eliminated. Obviously, the complete elimination of all errors may not be realistic, so some reasonable hypotheses could be made, taking into account that for some products and/or for some product families the adoption of a configurator may be unfeasible. The ultimate output of this phase is the identification of configurable product families. These product families are partitioned in groups characterized by similar commercial and technical models. For each group, a representative product family is identified. The remaining configurable product families within each group will be hereafter indicated as *non-representative, configurable, product families*.

Macro-analysis may be costly. Its cost depends on the number of product families the company produces, on product complexity, on how the customer influences the configuration process, and on how different the various product families are. A correct macro-analysis may take from one working week to a few months. How fast the macro-analysis is accomplished depends not only on the number of people involved, but also on the collaboration offered by the company and on the importance directors and managers give to the project.

10.4 Micro-analysis: evaluating implementation alternatives and costs

The micro-analysis is a detailed study of the current configuration process and of how it can be redefined with the support of a configurator. It differs from the macro-analysis because it reaches a

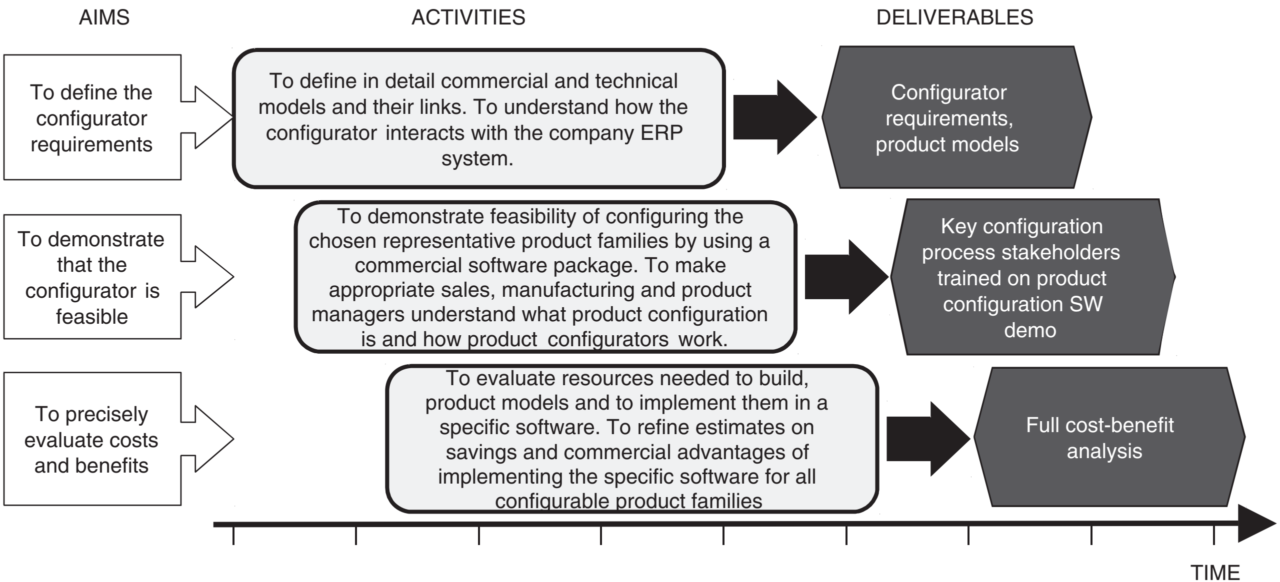


Figure 10.4: Micro-analysis: aims, activities and deliverables

high degree of detail and reliability in the evaluation of costs and benefits. Before starting micro-analysis, it is necessary to choose, from the priority lists developed during the macro-analysis, one or more product families representative of all the product families which are candidates to be supported by a configurator. Figure 10.4 shows aims, activities and deliverables of the micro-analysis.

Defining the configurator requirements. The first aim consists in defining the requirements the configurator has to satisfy. More precisely, it is necessary to take some decisions by answering a series of questions such as: How will the technical model and the sales dialogue be? What type of customer interface will be used? Will it be on the Web? Will the software be given to the customer or will it be kept by the salesman? etc. All these specifications will allow the formalization of the precise requirements of software vendors and the evaluation of whether their software is able to support the company's objectives.

For the chosen representative product families, it will be necessary to study, in detail, every single element that helps to define how the configuration process will be with the support of a configurator. The result of this study will be the detailed definition of the technical model, of the commercial model and of the links between the two of them. It is important to interview salesmen as well as technical and production office personnel. Furthermore, previous configurations must be analysed by studying commercial and production documents, such as bills of materials. In particular, the bills of materials of all the products belonging to the same family must be analysed, in order to understand at which level it is convenient to structure the generic bill of materials and how to obtain from this the other single bills. Also the coding rules and the existing codes must be analysed and, in the case of needed code modifications/changes, how to pass from the old codes to the new ones must be defined. Obviously, the changing of codes must be handled cautiously. Moreover, there must be a detailed definition of costing and pricing models. As far as sales dialogue is concerned, prototypes will be developed and the salesmen will decide whether they are acceptable or not. The result of this test is extremely important, otherwise we may run the risk of developing a tool that will not be employed by one of the most important users – the salesman. Product specifications collection, pricing, costing, coding, and the generation of bills of materials may be performed in different companies according to more or less strict rules and procedures. In the case of slack or unclear rules this phase may become an occasion for reorganization and rationalization.

Demonstrating configurator feasibility. The aim of this phase is to demonstrate that the configurator is feasible from the technical and managerial standpoint. Potential users may have some reasonable doubts about the quantity and complexity of the activities that the configurator is supposed to carry out. Therefore, it is important to demonstrate how an information system that fulfils these functions is not only technically practicable but also compatible, after the necessary adjustments, with the system adopted by the company.

During the micro-analysis, there should be a demonstration using some product configuration software. This demo takes place once the company has already started considering some software packages. Its aim is to assess whether the software is able to meet the company's requirements. It also provides precious inputs to the subsequent activity of cost and benefit analysis.

In order to assess configurator feasibility, the key company stakeholders, such as sales, manufacturing and product managers, have to be trained in the logic behind product configuration and in the characteristics of product configurators. A series of meetings will help them understand what product configuration may mean to the company and to tackle resistance, obstacles and misunderstandings. Very frequently, these meetings become occasions for analysing how the company is operating, because some issues that were overlooked are now discussed, such as: the overlapping of product families, the criteria used to compile bills of materials, the coding rules, the possibility of standardizing components, cycles, etc. Some of these problems must be solved before implementing the configurator; others, however, can be considered later, when thanks to the configurator, there is more order and more stability in the configuration process.

Evaluating costs and benefits with precision. The third aim of the micro-analysis is the accurate estimation of the time and resources necessary to implement the configurator. Once the two previous aims have been achieved, the company can precisely estimate costs and benefits for the representative product families. Based on these data, one can estimate costs and benefits of all the configurable product families similar to the representative ones more precisely. Obviously, the needed resources depend on the particular type of software used, because, for example, the way in which the constraints are described may be more or less user-friendly and so the efforts of the person who designs the technical model will be more or less effective. An important part of these resources is independent from the software and is devoted to collecting documentation, to analysing bills of material, to meetings, etc.

Micro-analysis may take a long time because it basically develops a prototype of the configuration system. Moreover, it has to be extremely accurate. Inaccurate micro-analysis may lead to selection of the wrong configurator or even to miscalculating the convenience (for the company) of implementing a product configuration system at all.

10.5 System design and implementation planning

This step involves deciding whether the product configuration system is to be implemented and, if so, how it is to be implemented. As the configuration system is made of both human and computing resources that contribute to accomplishing the configuration and modelling processes (see case 4.2), all these aspects have to be considered. Figure 10.5 shows the aims, activities and deliverables of this step.

Freezing the final product configuration process. Redesigning the configuration process means having a clear map of all the activities performed, from the collection of customer needs to the release of the product documentation necessary to produce any product variant (see case 2.1). Most importantly, the company should have a firm idea of which activities will be manned and which will be automated. In addition, each activity should be thoroughly described, as well as the information it exchanges with other related activities.

During this phase it is important to build consensus among managers and users about the redesigned configuration process. These decisions are made based on the results obtained from the analysis carried out in the preceding phase. If some competitors have already implemented or are implementing a product configuration system, it will be necessary to compare their decisions to those made by the company. Other ideas might come from other companies with similar configuration problems, even though they operate in different sectors. Consultants, advisers and software vendors can also help to obtain information about similar applications.

Choosing the software. The ultimate decision of whether or not to implement a product configuration system cannot be taken without choos-

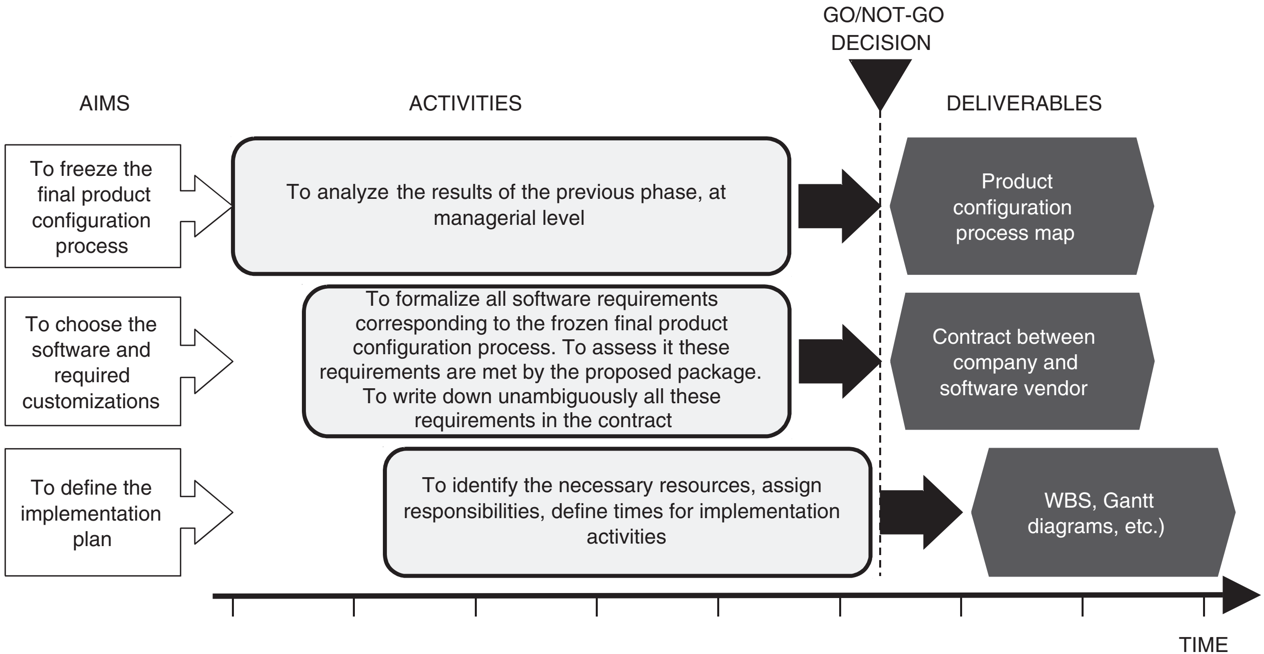


Figure 10.5: System design and implementation planning: aims, activities and deliverables

ing a software package. Product configurators, like many other software packages supporting a company's processes, have many different technical features (see Chapter 9). The higher the number of technical features, the more difficult it is to assess whether a software package is appropriate for a given user. Consequently, only if a company has understood configurator requirements in detail, during the micro-analysis, can it really assess whether a given package fits the company's needs. In addition, having detailed requirements helps the company to write a 'bomb proof' contract, thus reducing the risks of discovering software inadequacies too late, or spending a lot of money in software customizations.

In choosing the software, a common software selection must be followed. During this selection it is necessary to consider the specifications of the previous phases, which defined what the configurator is supposed to do, and are expressed by the technical features described in the preceding chapter.

There must be an accurate evaluation of the links between the configurator and the rest of the information system, particularly the parts described in Chapter 8. If a configurator is part of an integrated software system, it is important to control which modules of the system must be activated to obtain the desired configuration functions. In the case of stand-alone systems, it is essential to check the links to be established with production and sales modules, with the PDM, etc.

Defining the implementation plan. After choosing the software, the implementation plan must be defined in detail. Before implementing the configurator, it is necessary to time the implementation and identify and assign the required resources to the project. Such project planning activity has to be performed for each product family.

The micro-analysis provides precise information about the activities to be performed for the different product families, but it is still necessary to check implementation times. Therefore, it may be useful to run pilot projects to acquire some experience, to control timing of activities and to define homogeneous implementation criteria. If a company has similar product families, with analogous configuration problems, a single pilot project will be enough. Human resources needed for software implementation have to be identified. The directors of IT, sales and R&D have to be involved in this activity. It is important to determine the time each person will devote to this task and what specific responsibility will be assigned to each of them. For example, the technical office may devote a full-time and a part-time engineer to elaborate product models, while the sales office may devote a part-time domestic manager and a part-time export manager to define a sales dialogue appropriate for its users. Finally, the contribution to the implementation effort from the software vendor, in terms of programmers' and analysts' time should be included in the plan and agreed upon with the software vendor.

The length of the phase devoted to choosing the software and planning its implementation depends on the time taken by the negotiations and on the accuracy of the preceding steps. The decision-making process may be delayed if the results of the micro-analysis are imprecise and not convincing, or if there is internal resistance due to the fear of losing power or because of misunderstandings.

10.6 System implementation and launching

The final step involves all the activities required to have the product configuration system up and running for all the target product families. Although many conceptual hurdles, by this time, should have been surmounted, implementation and launching are not activities to be taken lightly. On the contrary, they imply an enormous work-load and require careful attention. Figure 10.6 shows a synthesis of aims, activities and deliverables related to this step.

To build product models for non-representative product families. Up until this phase, commercial and technical models have been developed only for those product families that have been considered in the macro-analysis as representative of all product families that are candidates for configuration. A fully fledged system implementation would also include such candidate product families. Even though the implementation for a certain family is very similar to that used for another family, it is always necessary to define constraints, codes, etc. for a product that is not exactly the same.

Debugging the system. Besides the implementation of dialogues, technical models, etc., system implementation includes a test on the correct functioning of the software and of the procedures that gear the computer system with the organization system. Firstly, the test is done on some 'ad hoc' cases by people working on the product configurator. This initial debugging phase has to be complemented by tests in real situations to verify that impossible configurations, wrong bills of

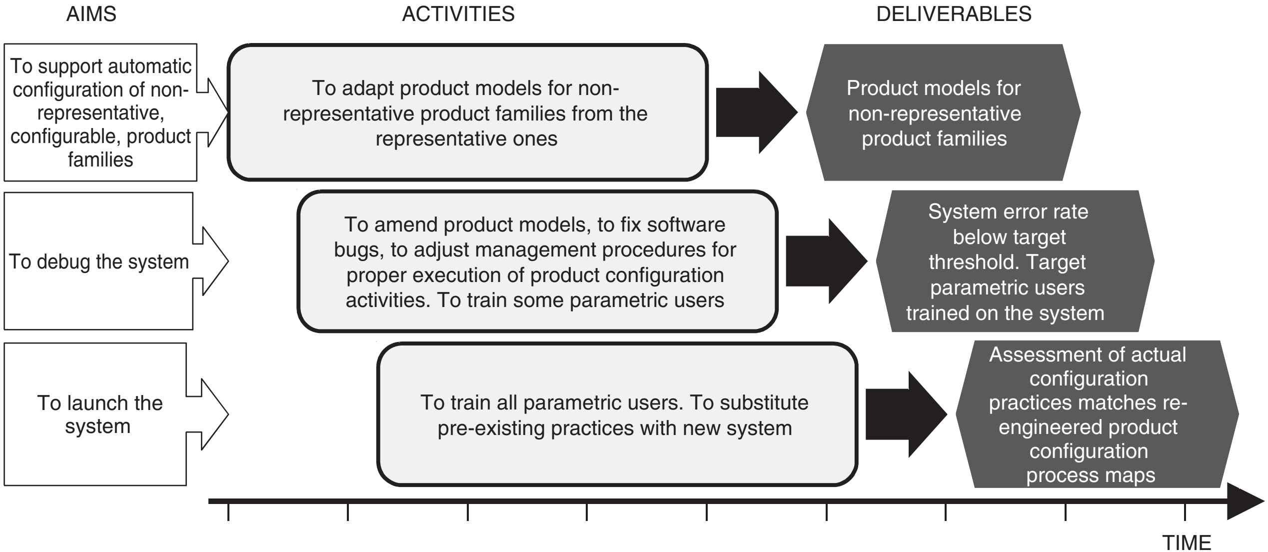


Figure 10.6: System implementation and launch: aims, activities and deliverables

materials for certain variants, etc. are not produced. For a period the configurator and the traditional process work in parallel, and the solutions obtained from both of them are compared. This ultimate debugging phase, besides identifying previously undetected errors and issues, is intended to demonstrate that the system works properly. System debugging may end when its error rate falls below a predetermined threshold.

Launching the system. Once the system has been sufficiently tested, it is definitively delivered to the different users and the traditional process is abandoned for the new one, supported by the configurator. Such substitution can take place in different ways: (1) the system is introduced for all the products and for all the users simultaneously; (2) at first the system is introduced for only one family and for a group of users and then extended to other families and other users. Some difficulties may appear at this point, difficulties related to the fact that some of the people involved in the process may not believe in the system and that those who use the system have to change some of their habits. This change in working practices requires a continuous support and monitoring of the users for an efficient application of the system.

The length of this step depends on the product complexity, the number of product families, the resources involved in the project and their responsibility, as well as the experience of the software vendor and many other factors. Therefore, the length can be extremely variable. Just to give an idea of a reasonable time needed for a typical project, we may say that the average duration is from a few months to a year. Longer times very frequently reveal the presence of some technical or organizational problems.

10.7 The elements of success

Simply following the steps illustrated in the previous sections is not enough to ensure the success of the implementation process. Although we have already pointed out that a balance of hardware, software and 'peopleware' is needed to meet the goal of efficient customization, here we want to emphasize once again the human dimension.

Management support is fundamental. Top management, in fact, should define a precise and appropriately communicated strategy in approaching the redesigning of the product configuration processes. The implementation of a product configuration system should be adequately sponsored, taking care that prospective users do not see the tool as a menace. Of course, top management has to clearly understand the initial costs of implementing the system and, in the case where the decision of implementation is taken, it should consider that cutting down initial analysis costs may compromise the whole project. Finally, a team of qualified people has to be gathered, including people from the different departments involved in the process. Even if external consultants or experts are also involved, to make up for the lack of expertise, the implementation will not be exclusively their responsibility and the internal personnel have to participate actively in the project.

The second fundamental component of the 'peopleware' side of the system is the personnel who are actually going to use it to configure products. Needless to say, salespeople are a fundamental component of the manned part of the configuration system. Six initiatives can specifically encourage these people.

First, involve personnel who have shown interest in using the configurator since the beginning of the project. The company must emphasize the fact that the new tool will help salesmen do a better job with their customers, while feeling more gratified. In this way, two advantages can be achieved: (1) the personnel have time to get accustomed to the idea of using a computer, avoiding the risk that a strong impact may discourage the use of the system, and (2) the salesmen may offer their own experiences, thus building a really valid configuration system.

Second, define and test the sales dialogue together with the sales people. The accomplishment of this requirement ensures the building of a tool precisely adjusted to the sales process. Consequently, there is a drastic reduction in the number of tasks to be done, in adjusting the configurator to the salesmen's way of working after the implementation. Obviously, the salesman does not possess all the necessary knowledge to handle the software, so his work must be initially supported by qualified personnel.

Third, use prototypes and simulations. As already mentioned in the preceding paragraphs, it is important to start using the configurator gradually, beginning with a prototype and then extending the use to other functions and other product families. Simulations may also be very useful to test efficiency, in different situations that take place during the sales dialogue.

Fourth, simplify the system usage. If the system is simple, salesmen and also engineers maintaining technical models will use it regularly and efficiently. In order to simplify its use, the system has to be as similar as possible to the tools previously used, reproducing their general structure and language. As already mentioned in Chapter 4, the graphic interface facilitates the use of the configurator.

Fifth, automate selectively. It is not necessary to involve the complete sales personnel in the use of the system, at the same time. On the contrary, it is convenient to start with people who are capable of using information technology. In this way, it is possible to test, in detail, the efficiency of the configurator before extending its use to all the sales force.

Sixth, provide assistance. All the companies we observed that were successful in the implementation of a certain information tool had trained their personnel with particular care. Salesmen must know how to use the configurator competently. Moreover, it is essential to provide a support to solve problems and to answer questions, so, very frequently a qualified person is appointed to carry out this fundamental assistance.

10.8 Project 'killers'

The project for the implementation of a configuration system may be abandoned or may not fulfil its goals for a number of reasons. The study of a number of previous unsuccessful implementations highlighted that it is particularly important to pay attention to the project 'killers' described as follows.

Changes in the roles of personnel. The introduction of a configurator generates changes in the roles of technical and commercial personnel. Quite often, some of the people involved in the project are opposed to such changes. For example, the freedom of action of the commercial personnel is limited by the technical support introduced by the configurator, therefore the salesman cannot promise a certain solution without verifying its feasibility: this lack of control may generate resistance to change. Thanks to the configurator the salesman has information about costs, but he must carefully motivate possible low margins. In other words, he possesses a more powerful tool but, at the same time, he must be more responsible. The technical office personnel, on the other hand, must formalize and share their knowledge. Those who used their knowledge to acquire a certain power within the company have to decide whether to accept the new help and lose part of their power or to boycott the project.

Poor inter-functional collaboration. The natural inter-functional conflict between front-office and back-office may become an obstacle when defining some of the preferences required by the configurator. For instance, when deciding about the number of colours and types of dye to offer for a certain product and how to price it, most probably the sales office and production will take contrasting positions. The former will underline the importance of offering multiple solutions to satisfy customer needs, while the latter will underline the costs and problems related to dyes and colours rarely requested. These different positions may create a good occasion to take reasonable decisions and the implementation process may proceed without great obstacles. On the other hand, if the differences prevail uncoordinated or incoherent choices may be embedded into the software and will obviously increase implementation complexity. In the worse possible scenario, such contrasting views cannot be reconciled. The project, then, will get stuck.

Excessive workload. Building a configurator requires a great amount of work and the collaboration of expert people. The normal activities of the company cannot be stopped in order to implement the configurator. Therefore, those who work on the implementation will be forced to divide their time between daily routine and the project. This problem is particularly important for high-level managers and for the experts involved in the project, because their time is always very limited. That is why the project sometimes does not make headway until somebody is relieved from their routine tasks and starts working full-time on the implementation.

Unreasonable architecture of product families. In many companies the different product families evolve causing many overlaps, without taking advantage of the similarities of some parts or modules. When implementing the configurator such situations arise clearly, and very often the management has to take decisions about rationalizing the offer and/or making better use of the various similarities. If the company decides to improve the offer and the use of similarities, the project field is expanded and consequently the series of new activities delay the implementation, running the risk of blocking the whole project. In addition, building a configuration system on product families with unnecessary complexity and variable architectures becomes an extremely hard task, not only for the implementation but also for the system updating.

Excessive software customization. When software requires extensive customization to fit the company requirements, implementation times tend to escalate as well as software costs. As a result, it will be necessary to refinance the project budget, without obtaining immediate results. This situation obviously discourages businessmen and they may decide to interrupt the project.

Part IV

Operational and Organizational Implications of a Configuration System

11
A Comprehensive View: the Sideco Case Study

In the previous chapters we discussed various specific aspects related to product configuration: the problem of product configuration, configurable products, possible approaches to product configuration, etc. Each aspect was illustrated with examples, but up to this point, we have not considered a comprehensive analysis of all the aspects associated with redesigning the configuration system in a company that intends to improve competitiveness. This chapter presents a complete example of how one company effectively managed to redesign its configuration system. In particular, this chapter deals with:

- the original configuration process and the problems the company had to face;
- the project for implementing the new configuration system;
- the solutions used for product modelling;
- the new configuration process and its advantages.

11.1 Product and company context

Sideco S.p.A. is a small manufacturing enterprise which employs about fifty people and has revenues of more than 5 million Euros. The company has reached an important position thanks to its innovative policy. Recently, with the introduction of a new quality control system, the company was granted the ISO 9002 standard. It also implemented an advanced configuration system that will be discussed in this chapter.

The enterprise manufactures mould-bases for plastic moulding and punching-bases for metal sheet punching. The mould-bases for plastic moulding represent the main product of the company, as they account for 60–70 per cent of sales. Specifically, Sideco's focus is on medium and large sized customers, working for the automobile, home appliances and sports footwear industries. Mould-bases are either normalized or made to the customer's own design, therefore there is little margin for differentiating its products from those of its competitors. Low product differentiation, in turn, led to a general alignment of prices among competitors. One of the main ways to differentiate products is the level of service, i.e. the capacity of the company to deliver rapidly the needed solution to customers. This aspect is even more critical if we consider that a portion of the sales comes from spare parts: the cost of a mould-block has marginal importance compared to the cost of the production stoppage caused by mould-block failure. Delivery speed, therefore, is an order-winning criterion in this niche business.

The mould is needed to connect the die to the press and, therefore, it works as an interface between a highly variable object (the die) and a fixed one (the press). The mould-block is essentially made of a set of steel plates that are connected through guide pillars that slide into holes drilled on the plate. Other complementary accessories are used, such as screws, ejector pins, springs, etc. (see Figure 11.1).

The mould-block is a highly configurable product. The various plates can have different sizes and thicknesses; be combined in different ways; be made of different types of steel; be connected by different sliding pillars and be passed by different pins, requiring different holes to be drilled and different bushes to be inserted in the holes; and they may require special features ordered by the final customer.

The logic followed to guarantee the product variants is shown in Figure 11.2. For example, variants 1 and 2 share the same upper plate,

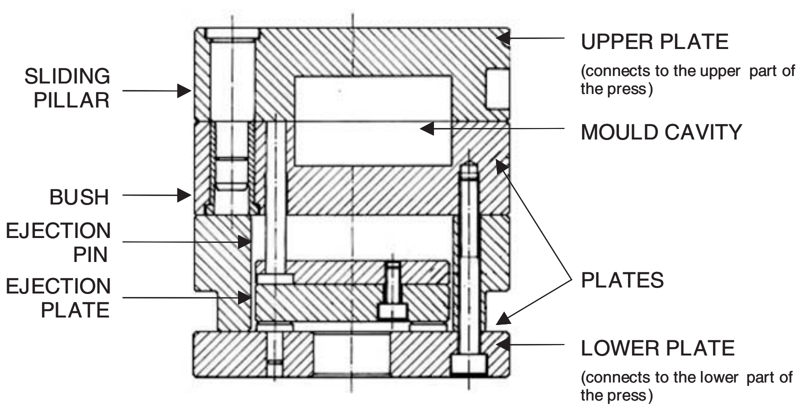


Figure 11.1: An example of mould-block

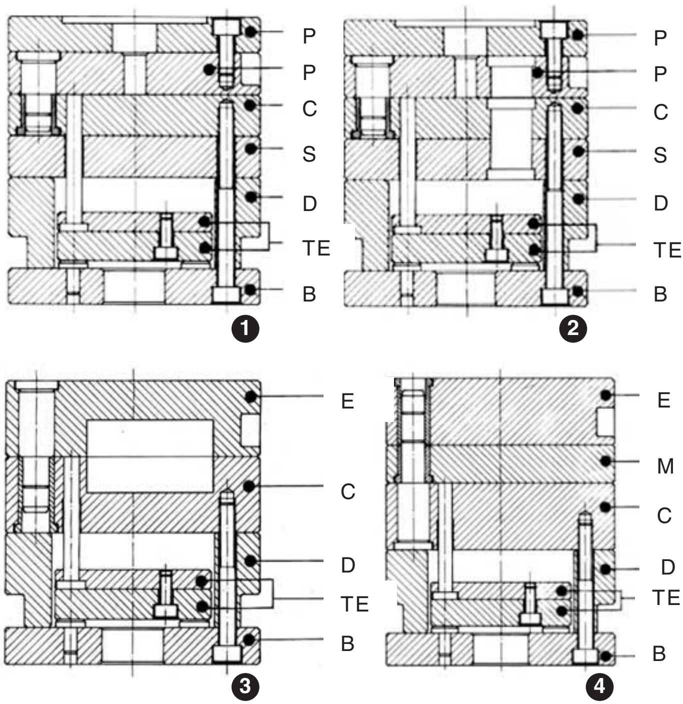


Figure 11.2: Examples of different mould-block configurations

lower plate and distance pieces (B, PC and D), but differ in the intermediate plate (Ps, C and S). However, a more accurate analysis shows that the plates differ one from the other only by the presence of a hole. The same conclusion may be drawn from variants 3 and 4. These parameters generate a huge variety of parts, which is mirrored in the more than 1000 finished product codes that are currently handled by the information system.

To control the complexity induced by the proliferation of final product configuration in the operational environment, the companies in this business introduced the concept of normalized mould-blocks. This entails the normalization of the possible dimensions (height, width and depth) of the plates into a fixed size range, as well as the normalization of the possible diameters, length and shape of pillars and other sliding parts. However, the possible combination of parts remains high. Moreover, often the customer requires products with specifications that fall outside the normalized size range, further increasing product variety.

11.2 The original configuration process

The following paragraphs provide a description of the activities triggered by a customer bid and then by the order entry. The aim of this description (see Figure 11.3) is to portray how product variety affects part of the company's operations, rather than to provide a detailed snapshot of the tendering and order fulfilment processes.

When a customer submitted a bid to the company, he referred to its catalogue, first identifying a family. Then, within this family, the customer defined the exact specifications of the needed product. In certain cases, for example, the customer needed products that did not exactly conform to a given product family because of the different sequence of plates or because of specific plate sizes, etc. If the product was a 'standard' product, the customer picked the code from the catalogue or could obtain the exact code by following simple rules defined in the catalogue. In the bid, the customer included both the written product specifications and the product code. If it was a 'custom' product, the customer had to include in the bid a technical drawing, so that the company could have enough information to prepare the tender. After receiving the bid, different activities were performed in the technical

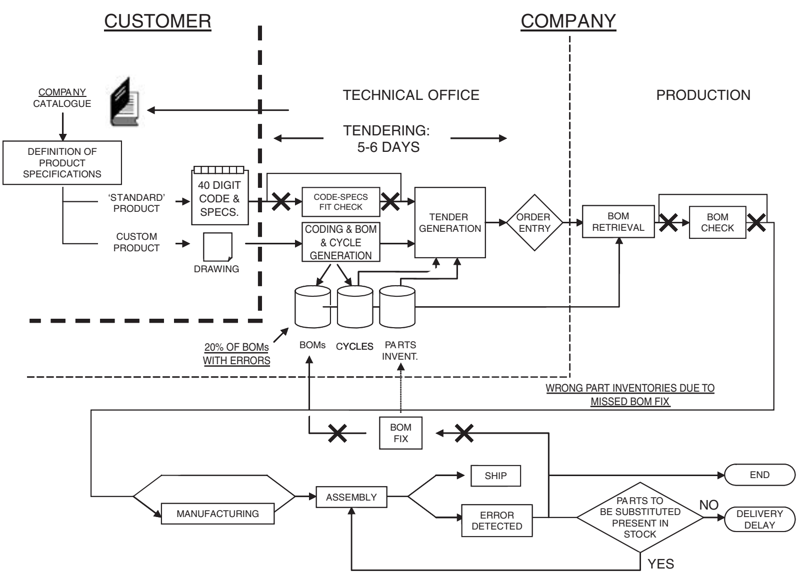


Figure 11.3: The configuration process before the introduction of a product configurator

office if a 'standard', rather than a 'custom' product was requested. In the case of 'standard' products, the fit between product code and product specifications was checked, in order to avoid possible errors in coding or in reading the catalogue. In the case of 'custom' products, the technical office generated the product code, based on the drawings received, and also created the product BOM and production cycle. This activity was time consuming, and it could take from twenty minutes to one hour for the most complex products, and every day the technical office had to process dozens of bids. For this reason, the checks on the 'standard' product bids tended to be bypassed, as the personnel concentrated on the more demanding task of generating product data for 'custom' products. Once all the information needed to make a tender (BOM, manufacturing cycles, parts inventories, etc.) was available, the product could be priced and a delivery date (basically dependent upon the presence of the required materials in stock) could be defined. The average lead-time for an offer was about five to six days.

Once the order was received, the responsibility for producing and shipping the product passed to the production area where the production manager first retrieved and checked the correctness of the BOM. Indeed, due to the many tasks competing for the production manager's time, only rarely was he able to check if the BOM was correct, as he also had to take care of CNC machine programming (whenever manufacturing activities were needed) and scheduling shopfloor activities. Based on the BOM and inventories, he issued possible purchase orders to suppliers and released the work order on the shop floor. Due to missing BOM checks, errors were often detected only during the assembly operations by workers. If the wrong parts were currently in stock there were no serious consequences on delivery time, as it was possible to immediately get them from the inventory, perform the needed operations and assemble them with the other parts. In contrast, when the wrong parts needed to be substituted by parts that had to be machined or even purchased, then a delivery delay was very likely. According to the company's procedures, whenever an error was detected in the assembly process, the BOM should have been corrected. However, this feedback loop always failed, although for trivial reasons. For example, the workers had not had time to make the corrections because they were working on the final assembly in order to meet the carrier pick-up deadline. In other cases, the personnel in the technical office might not have noticed the correction in the BOM because it had been written in the same colour as the printer ink! Consequently, the company reached an error rate in its BOM of approximately 20 per cent. The company, under the pressure of delivery times, was forced to bypass a series of checks that caused errors, which in turn had a negative influence on the capacity to deliver on time. These delays also affected the company's competitiveness. The apparently obvious solution of increasing personnel, in order to carry out all the necessary checks, was not feasible because the product cost would have increased excessively. Unfortunately, it was not a reasonable alternative, due to the fact that in a market where many competitors offer the same product, the price was and still is fixed. In synthesis, the enterprise faced a paradox: since time was a vital competitive factor, it was forced to reduce and even bypass some activities, introducing errors that ultimately had negative influences on delivery times.

11.3 The solution: a software to support product configuration

As mentioned in the previous paragraph, the main problem the company is facing is that of *ensuring the correctness of product information without reducing customization* (something impossible due to specific and heterogeneous customer needs) and *without increasing control costs excessively*. The basic idea that inspired the company's approach to the solution of this problem was to automate check activities in the process of product information generation. Given this specific need, the company decided to implement a product configuration software, i.e. a software tool that is capable of 'translating' the customer specifications into all the product information that is needed for tender generation and product manufacturing. Specifically, the company implemented a stand-alone product configuration package (Galileo TCE Tender Configuration Engine) that is developed and installed by a software house (San Marco Informatica) specialized in serving small and medium-sized enterprises. Since a detailed description of the implemented software is not the main purpose of this book, we will focus in the following paragraphs on the way the product configuration software has been implemented and its impacts on the company's operational performances.

11.4 Product modelling

The main issue in implementing the product configuration software has been product modelling. As mentioned in section 11.1, the multiplicity of features that can be customer-defined leads to very high product variability. As product variability is not the outcome of irrational customer decisions, but arises from customers' technological needs, then the product model has to adapt to it. A highly variable product model tends to be complex, especially when there are many interdependencies among product characteristics. For example, an ejector pin requires holes of appropriate size to be bored through all the plates that the pin has to pass. Modelling complexity is exacerbated by the fact that a really useful product model has to embrace all the product families, so that the customer could be easily driven in the selection of the product family that best fits his needs. The difficulty of capturing the company's product knowledge in terms of rules and formulas, initially slowed the project, leading deadlines for the release of a working product model to slip from three months to one year. The project did not really take off until an employee was relieved from his daily activities and committed full-time to the implementation of the product model.

A further important source of complexity in product modelling lies in the fact that customer specifications (e.g. position of the holes, complexity of added manufacturing activities, etc.) not only affect assembly, but also manufacturing activities. For this reason, the production cycle has to be included in the product model, so that it can be dynamically generated based on customer specifications. The product model, in the end, allows for the definition of the BOM and production cycles. These two pieces of information are of capital importance in the tendering process, as they allow product pricing to be defined.

Product data (BOM and production cycles) are obtained by *relating* the generic bill of materials (see Figure 6.13) to the configuration dialogue, for example to the set of questions the customer has to answer in order to describe product specifications. This set of questions leads to the generation of the product code that has been reduced to 13 digits, 9 for the family and 4 for the specific variant (the digits are assigned sequentially). Once the company implemented the product model and the related routines for product pricing and coding, it became evident that the work done could easily be used to implement a further feature: the automatic generation of a two-dimensional CAD drawing by means of a *parametric model* (see Figure 11.4).

11.5 The new configuration process

The definition of a *sales dialogue*, its links with the product model in order to obtain the BOM, the production cycle and then the product

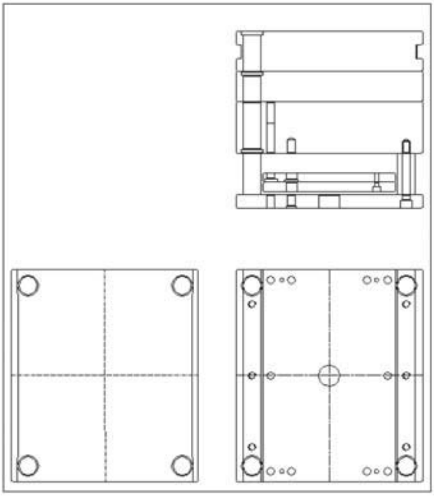


Figure 11.4: Graphic output of the configuration system

price, as well as the implementation of a parametric model for automatic drafting, have important operational and strategic implications that affect both the company and the coordination between the company and its customers.

As the company's internal processes were affected, the implementation of the product configuration software resulted in two different kinds of advantages:

- reduction of manned activities in the tendering process (tendering lead-time from 5–6 days to 1 day);
- increase in the level of correctness of product information (almost 100 per cent)

Concerning the reduction of manned activities in the tendering process, this is an obvious consequence of the presence of product models that automatically generate product information. Especially in the case of 'custom' products, the coding as well as the BOM and production cycle definition activities disappear. Also, an increased level of correctness of product information eliminates the need to perform certain checking activities, such as the product code check, the product specifications fit check in the tendering process or the BOM check before production starts. Since these activities were often bypassed because of haste, the more correct product information results in fewer errors being detected in the assembly phase and consequently in greater delivery timeliness.

In addition, the company gave its customers an executable CD-ROM version of product configuration software. This application:

- guides the customer in defining the product specification through a configuration dialogue;
- gives the customer flexibility in defining specific product variants (for example, different plates assembly schemes);
- generates a two-dimensional CAD drawing of the mould-block that corresponds to the customer's specifications.

The advantages related to the release of a CD-ROM version of product configuration software to customers can be seen in terms of operational results and in terms of satisfaction and creation of durable relationships. From the operational point of view, the customer started using the software developed by the company and his product coding, specification descriptions and product drawings are always expressed using the conventions set up by the company. This in turn reduces the possible distortions in the company–customer communication channel, diminishing the chance of delivering a product that does not conform to customer needs. At the same time, through the product configuration the company drives the customer to order objects that fall into its normalized product range. In fact, the software tells the customer when he is defining specifications that fall out of the normalized product range and that, therefore, are likely to require longer lead-times and/or increased costs. Reducing the incidence of products out of the normalized range allows for greater efficiency in production.

Considering, finally, the strategic implications of the adoption of the product configuration system, it is important to recognize that it represents a way to tie the customer to the company by offering a service that is specially designed to improve the coordination between the company and its customers in the bidding-tendering process. The information that is released by the software is tailored to the company's internal products and language. Product description, drawings and codes are all company-specific and, therefore, cannot be used by competitors. The pay-off for the customers, besides the positive effect of better coordination, is the reduced time in generating product specifications and drawings.

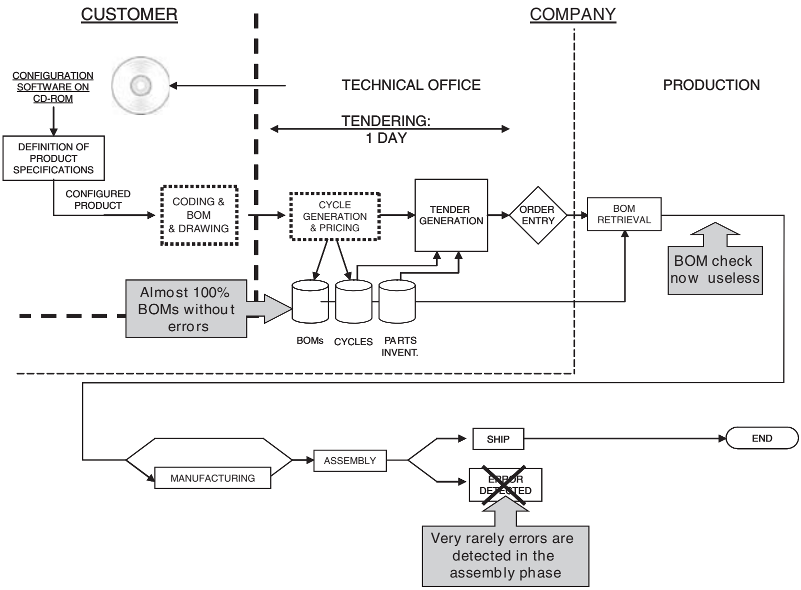


Figure 11.5: The configuration process after introducing the product configurator

Figure 11.5 illustrates how the effects of product configuration software implementation propagate to parts of the company not directly involved in the implementation, such as the production department. Moreover, this case study demonstrates that the changes in the workflow induced by the implementation of a product configurator are not necessarily confined within the company boundaries, as they can affect also the customers' organizations. The product configurator, therefore, can be seen as tool that does not just improve internal working practices, but that improves inter-company coordination by aligning working practices between the companies involved in the bidding-tendering process.

11.6 A special case?

This case study highlights the importance of considering the configuration process as a whole, in order to understand where and how it may be improved. However, the Sideco case has a double interpretation. On the one hand, as we mentioned at the beginning of this chapter, it presents a series of information that helps to 'put together the pieces of the puzzle', understanding how the different concepts discussed in the preceding chapters are combined in a real case. On the other hand, the case conceals a series of more general messages about the organizational activities a company has to undertake to get benefits from the product configuration logic. The Sideco case is 'special' because the company managed to take all the fundamental steps to reach its objectives, but at the same time it is a 'general' case, because it provides a lesson on how to reorganize the configuration system in a company. This is the subject of the final chapter.

12
Configurational Approach: Aligning Product, Processes and Organization Systems

Configuration systems have been defined as socio-technical systems, characterized by a technical component – the configurator – and an organizational component – people, procedures, processes, etc. To improve configuration system performance it is necessary to act on both technical and organizational dimensions. This chapter explains the logic used to maximize product configuration process performance. In particular, it:

- illustrates how automation of repetitive check activities allows the 'delay syndrome' and sluggish red-tape to be surmounted;
- highlights the changes in the roles and tasks of salesmen and designers, necessary to deploy the potential advantages of configurable products;
- explains how the configuration system may work as a mechanism for inter-company coordination;
- proposes an integrative model of the technical and organizational actions needed to efficiently offer customization through product configuration.

12.1 Overcoming the 'delay syndrome' and sluggish red-tape

The case study described in the preceding chapter clearly illustrates a problem that is very frequently faced by enterprises that offer configurable products. This problem is so widespread that it can be considered a real 'syndrome'. It originates from the need to speed up the order acquisition and fulfilment process. Due to this pressure on timing, many configuration checking activities are rushed through, or are skipped altogether. Poor or missing quality checks lead to a series of errors that, in the end, delay delivery times. In an attempt to meet the promised delivery date, the company is forced to take short-cuts which, paradoxically, instead of reducing time increase it, and the company is not able to meet the promised delivery dates.

The dynamics of 'delay syndrome' illustrated in Figure 12.1 can be found, above all, in small companies where configuration errors can be systematically corrected after they have been detected. This approach cannot be followed in the case of medium and large companies, where configuration errors have to be intercepted at the very beginning of the process. In these companies, in fact, to solve configuration errors late in the order fulfilment process, entails excessively high coordination costs. Duly, these companies use specific procedures to check order correctness and feasibility at an inter-departmental level. The different departments check the feasibility of the proposed configuration from their respective points of view: technical features, quality-related aspects, materials, delivery dates, etc. The most frequent method consists in grouping offers into batches (on a daily, or twice-day basis, etc.) in the commercial office, where some employees are in charge of compiling offers. Afterwards, this 'batch of offers' travels along product development, purchasing, quality assurance, production planning, administration, etc. Each person in charge of checking might make modifications

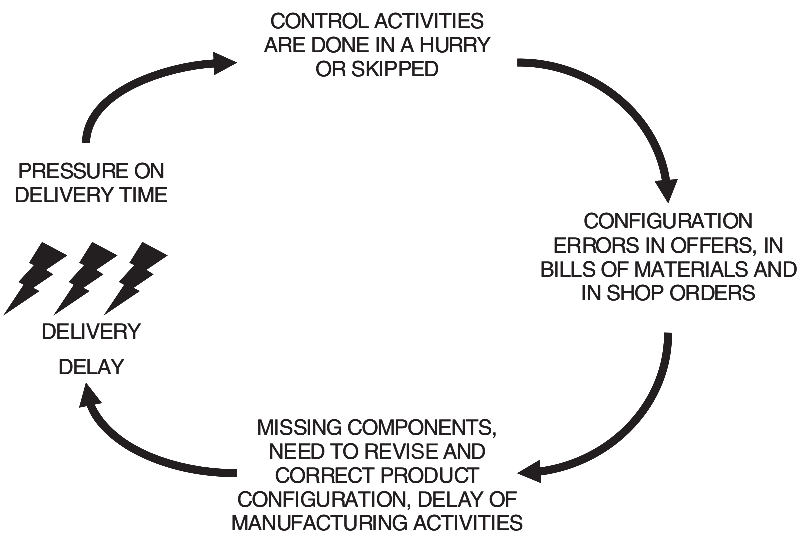


Figure 12.1: Delay syndrome dynamics

and comments. Finally, the offer returns to the sales office (see Figure 12.2). This cycle increases the time the company takes to get back to its customers, for at least three different reasons. First, checking activities are performed following a 'batch' logic. Although the examination of a configuration may take, let us say, ten minutes, the cycle in the technical office could take a whole day, because the usual procedure may be to control the offers in the afternoon, on a daily basis. If this practice is repeated in each office, the whole order cycle will take several days or even one or two weeks. Consequently, the company's capacity to respond promptly to customer requests will be reduced. It must be noticed that the 'batch' approach, in itself, is not incorrect. Whenever an employee examines an offer, leaving aside another task, there is a time-consuming mental adjustment. He has to postpone his current activity to start controlling the orders, concentrating on different details. Obviously, if the controls were made 'order by order', the employee's efficiency would be drastically reduced. The only exceptions to these problems are the companies with a very low number of orders per week, because the syndrome does not really affect them.

It does not matter whether a company is small or large, product customization will tend to inflate delivery times, either because of the delay syndrome, or because of cycle time lengthening. In both cases a lot of the checking related to product configuration is repetitive. As

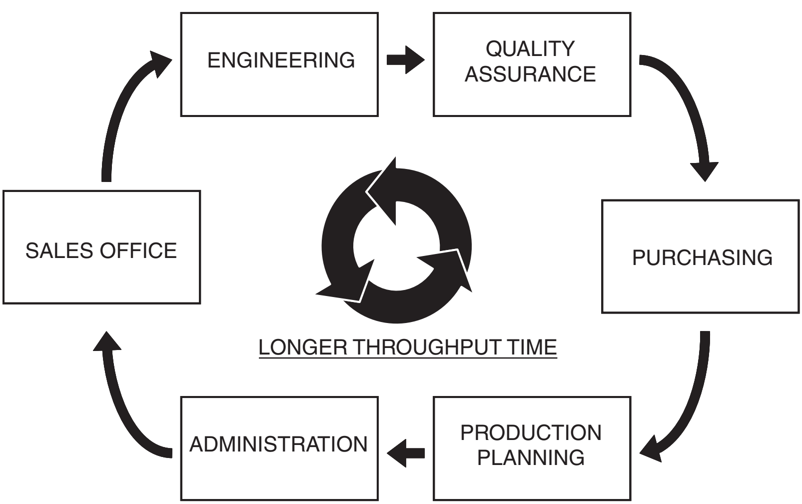


Figure 12.2: Cycle time expansion while checking the order

such, it is an ideal candidate to be translated into algorithms and then automated. Automation cuts errors in the configuration process, virtually zeroing fire-fighting activities and, ultimately, reducing the delivery time. Likewise, automation speeds up checking activities, possibly skipping some departments, so that the order processing cycle time is cut. For these reasons, configuration systems positively impact on the time a company takes to bring its customized products to the market.

12.2 Changes in roles and responsibilities

The decision to configure the product rather than to design 'ad hoc' variants for specific customers, has some important consequences for the roles and tasks of salesmen and designers (Figure 12.3). First of all, we must remember that when a product is configured, starting from a completely formalized architecture (see case 2.2) each variant is obtained by combining predefined components (standard, standard with options or parametric). This means that we can *define 'a priori' the possible configurations* (see case 2.3) by establishing: (1) different ranges of variation admitted for the various product commercial characteristics or attributes and (2) the links between such attributes. The definition of rules to identify feasible combinations of commercial characteristics allows the commercial configuration task to be assigned exclusively to the front-office, without involving the back-office, i.e. the production and technical office.

If the continuous interaction with the technical office is eliminated, customer requests will be quickly answered, offering a better service

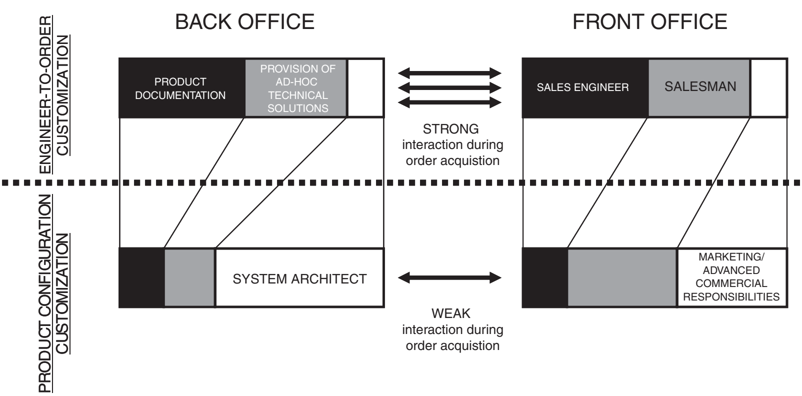


Figure 12.3: Changes in roles and responsibilities

and reducing costs. The roles of sales office personnel change considerably, because the room for negotiating product features with the customer is determined by the product space (see case 2.3). Such space is limited, if compared to a simple sheet of paper, where the salesman can freely write down 'what the customer wants'. In the case of configurable products, the salesman has precise indications that constrain the solutions he can propose. However, at the same time, he can assure high customization and, as there are no interactions with the back-office, he can proceed quickly and beat competitors. In addition, the time saved by the salesman in understanding what the company is able to do in order to satisfy customer needs, can be devoted to meeting other customers, analysing what competitors are doing, studying the market, in other words doing his job!

Also in the back-office the role of the designer is likely to change. He is less involved in the order acquisition process. His key competence will not be to find a slightly different solution each time for each customer or to painfully retrieve previous configurations. Rather, he may devote his best skills to creating the complete architecture of product families the company intends to offer. It is not an irrelevant change in the designer's tasks: instead of optimizing the single variant he has to optimize the product family! The mix of skills is modified: now, he must not only identify precise technical solutions but also operate as a 'system architect'. His responsibilities evolve towards the disciplined planning of product variety.

12.3 A new knowledge management process

The implementation of a configuration system supported by a configurator allows the reduction of the degree of interaction between front- and back-office. In this way, each office can focus on its specific functions. It could be objected that such functional focus is opposed to the concepts of process management and inter-functional coordination prevailing in management manuals and in the suggestions of business consultants. This is not true for two different reasons.

First of all, the need for inter-functional coordination during product configuration disappears. To understand why it is possible to decouple product configuration activities of front-office and back-office, it is necessary to go back to the very idea of a configurable product. Configurable products allow the deterministic relating of a set of predefined product attributes to product components (see case 2.3). Configurators exploit this deterministic correspondence to automatically translate product attributes into product components. Since this translation activity is typically done by back-office personnel, product configuration can be ultimately performed without involving them. Coordination between product configuration activities is still there – it is merely embedded in the machine!

Secondly, annihilating inter-functional coordination during product configuration is not freely acquired. Setting-up and maintaining the product configurator requires a new knowledge management process. As discussed in Chapter 10, building commercial and technical models (see case 4.1), as well as linking them, require an intense collaboration between front-office and back-office. This collaboration takes place during the *modelling process* (see case 4.2). The modelling process is a particular knowledge management process feeding the *product configuration process* (see case 2.1). Its input is the product knowledge tacitly present in the mind of technicians and salespeople and, partially, in algorithms, technical-commercial documentation and other available material the company may have. Its output is represented by the technical and commercial models (A) and by their corresponding documentation (B), as indicated in Figure 12.4.

When a company offers customization through configurable products, it reduces the degree of interaction between front- and back-office in the product configuration process. At the same time, it is necessary to *define a new process: the modelling process*, which, on the contrary, implies a high degree of interaction between front- and back-office. Therefore, the collaboration between front- and back-offices does not cease when an automated configuration system is implemented. The innovation arises from the fact that the interaction does not take place during the order acquisition and fulfilment process, but during a

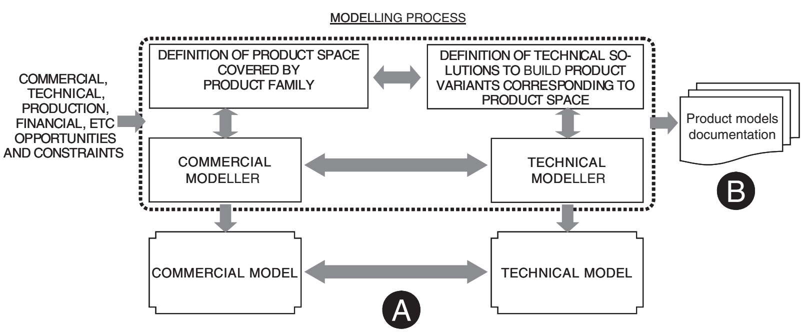


Figure 12.4: Modelling process

specific knowledge management process, called the product modelling process.

It is important to remember that from the technical and commercial point of view the output a modelling process must essentially produce is the necessary product model documentation. The quick modifications of such models and the frequent personnel turnover in key positions within the company, make it necessary to have information about how the product was represented within the configuration system. This information helps to transfer the responsibility of managing the system to new personnel and facilitates system updating. Case 12.1 illustrates these problems.

Case 12.1 The lack of system documentation: the case of a wood working machinery producer

Macclegno is a company that manufactures small machines for wood working: circular saws, planers, etc. The company exports 85 per cent of its production, with a great proliferation of variants according to the country, especially related to different electrical standards. Configuration management is particularly important for Macclegno, since it manufactures durable products. In case of machinery breakdown, it is necessary to immediately identify the specific components of the broken machine, in order to repair it as soon as possible, because it is blocking production. Macclegno implemented a successful product configurator, which supports around 80 per cent of the configurations. The problem this company had to face was due to the resignation of the only person who handled the configurator implementation. His work had not been documented and the configurator did not support product model documentation. The company was not able to complete the remaining 20 per cent of the configurator project. In addition, as the system was still in the debugging phase, the continuous corrections made by the 'system's master' were missing. Finally, nobody knew how to handle the technical and commercial model and so their updating was interrupted. In synthesis, the lack of system documentation and the incomplete structure of the modelling process caused a fatal blockage in the configuration system.

12.4 Different order acquisition and fulfilment processes

The organization changes discussed up to now are closely related to the presence of configurable products and to the implementation of information tools that support configuration. However, a product family is not always completely configurable (see case 2.4): a certain part of the product may be configurable while the other part is engineered to order. In this case we speak of *partial configuration*. Very often, within a certain product family there is a configurable 'sub-family', while the rest is not configurable. This happens, for example, when market considerations do not allow the necessary limitations to be placed on the customer to maintain the configurable product. In other cases, a situation of partial configuration occurs because the investment needed to redesign the product family and make it completely configurable is extremely high.

The possible simultaneous presence of configurable, non-configurable and partially configurable products must be taken into account when defining product management methods. Frequently, in this case, it is convenient to adopt order acquisition and fulfilment processes differentiated according to the type of product. If a unified process were adopted, many benefits would be lost because the documents for configurable products would follow the same itinerary as those for 'engineered to order' products. This would have negative consequences on times and efficiency during the order process.

The solution of a unified process for both types of product – configurable and non-configurable – is frequently used. In the absence of an information system that supports configuration, this solution has some advantages, in terms of simplification, for the whole organization. As soon as some activities 'disappear', because they are automated, the convenience and – in part – the possibility of using a unified process quickly diminish.

Case 12.2 Configurable and special products: the case of electric motors

Rotamax is an established medium-sized company specialized in manufacturing and selling low and medium voltage electric motors, DC motors and synchronous generators. Figure 12.5 shows the results of a study on the feasible configuration of asynchronous motors.

Broadly policode, the offer of asynchronous motors can be divided into (1) standard, (2) 'transformed', (3) special and (4) configurable motors. *Standard* motors are made to stock without any configuration. *Transformed* motors are essentially equal to standard motors kept in the warehouse. When they are needed, standard motors are picked from the warehouse and a limited number of components are substituted. *Special* motors are made to order and require 'ad hoc' design, codification and documentation. Finally, *configurable* motors are variants included in catalogues, therefore they are pre-codified, but not necessarily engineered. Some electrical parameters are particularly defined for these motors, such as number of poles, frequency, power, assembling, input voltage, as well as some accessories such as terminal box, shield, fan cover, junction box, etc. In the case of configurable motors, the configurator (Baan Configurator) is able to specify the bill of materials and create the cycle, without involving the technical office. Technical assistance is required in the case of extremely complex variants ('out of structure') with manufacturing activities that are not codified, therefore such activities must be specifically determined for each single case.

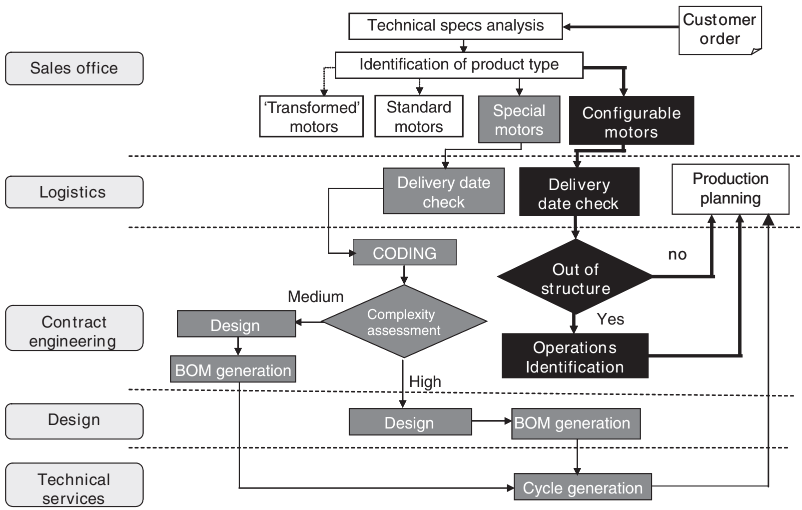


Figure 12.5: Different processes for configurable and special motors

This case demonstrates that configurable and non-configurable products follow different processes. Moreover, Figure 12.5 shows that completely configurable products and partially configurable products (the so-called 'out of structure' motors) follow partially different processes. Such process divergence eases cost accounting. In fact, it becomes easier and more reliable to calculate product customization cost for completely configurable and partially configurable products. Accordingly, it may be easier to compute margins of engineered-to-order products.

12.5 Inter-company coordination

The possibility of decoupling commercial configuration and technical configuration not only reduces coordination costs within the company during the configuration process: as illustrated in the Sideco case, the possibility of carrying out commercial configuration independently from technical configuration permits commercial configuration tasks to be assigned directly to the customer. This means that a whole portion of the configuration process takes place outside the company, and consequently operational processes are simplified. At the same time, this transfer of activities, from the enterprise to the customer, may be seen by the latter as a service, because he is able to simulate, in the ways and times he prefers, alternative product configurations.

The potential impact of product configuration on inter-company relationships is not reduced to the separation of commercial configuration activities. Let us consider the following situation: an enterprise buys a product component that can be supplied in numerous variants, for instance, a compressor for an air-conditioning system. In this case, to include in the technical model even the configuration of the component to be purchased, gives the opportunity to request, in a fast and precise way, what the supplier has to produce and deliver.

Considering the potential advantages in the interaction between the customer and the supplier, we may conclude that the configuration system is not necessarily limited to enterprise boundaries, but can also be a mechanism to improve inter-company coordination in the presence of high product variety.

Case 12.3 Buyer–supplier coordination and configuration system: the case of dashboards for industrial trucks

Plastikona is a company that produces dashboards and supplies one of the most important European commercial and industrial vehicle manufacturers, Mulotruck. The company has obtained a strong position in the market by

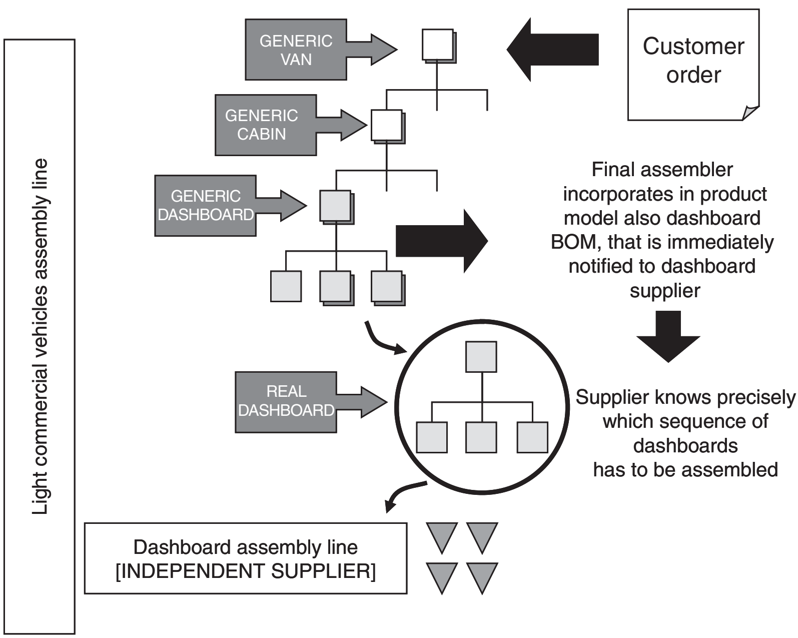


Figure 12.6: Enterprise-supplier coordination through technical model

offering high quality moulded dashboards and a complete service, which includes engineering and industrialization processes. Following the process of rationalization adopted by the transport vehicles industry, Mulotruck aims at obtaining from Plastikona not only the moulded dashboard, but also the complete dashboard assembled with warning lights, switches, instruments, wiring, etc. Unfortunately, the different combinations of nearly 300 codes, needed to build such complete dashboards, increase the number of complete dashboards variants to around 1000. Obviously, it is impossible to have the assembled dashboards in stock, due to high product variability, yet Plastikona must be able to supply Mulotruck with the exact mix of dashboards requested as quickly as possible. The solution found was to include, in the product model built by Mulotruck to support technical configuration, the generic component 'assembled dashboard'. In this way, according to what the customer requested, Mulotruck is able to elaborate a complete bill of materials for Plastikona, specifying which variant is needed (see Figure 12.6).

Since Planstikona receives the bill of materials already completed, it is not necessary to waste a second to decide, each time, how the product must be built. The dashboard assembly line, within the industrial vehicles factory, is able to produce any type of assembled dashboard configuration in only forty-five minutes. As a result, the dashboard assembly line manages to supply the main vehicle assembly line with maximum flexibility, without generating rigidity in the factory master planning and at the same time eliminating risks of errors in the dashboard configuration.

12.6 Configurational approach

The previous paragraphs complete a general overview of the activities an enterprise has to undertake when it decides to offer customization through configurable products. These activities refer to changes along three different main lines: (1) product, (2) product knowledge management systems and (3) the organization sub-system that performs configuration and product modelling activities. To adopt the *configurational approach*, that is to say, to compete by customizing configurable products, means to carry out an integrated action on all these three main lines, as discussed hereafter.

Designing a configurable product. It is evident that the starting point is the product. The company is no longer limited to designing single product variants; on the contrary, it designs or periodically updates complete product families. To adopt the product family as a working unit, rather than the single variant, requires changing the attitude of the technical staff as well as of the marketing personnel. A configurable product family must not be understood as a simple set of variants, but as a group of components and links between components, which can be combined and/or dimensioned according to a series of attributes.

These attibutes can be given a formal representation: the product space.

Designing systems to manage product knowledge. The second fundamental main line is to upgrade the infrastructure the company uses to process and manage product knowledge. The combinatory and/or parametric logic underlying the generation of different variants in a configurable product family requires a considerable effort to formalize product technical-commercial knowledge. At the same time, the huge amount of information may be difficult to manage. A product configurator becomes an essential tool to gather product knowledge and to turn into algorithms the repetitive aspects of the configuration process. The design of the supporting system is not limited to the configurator. It must be linked to the Manufacturing Planning and Control System and possibly to Customer Relationship Management and Product Data Management systems.

Designing the suitable procedures, roles and responsibilities for offering customization through configurable products. Finally, the 'human' side of the system must be reconsidered. As we have already discussed in this chapter, this main line may entail multiple activities: the redefinition of roles and responsibilities in the technical and commercial personnel,

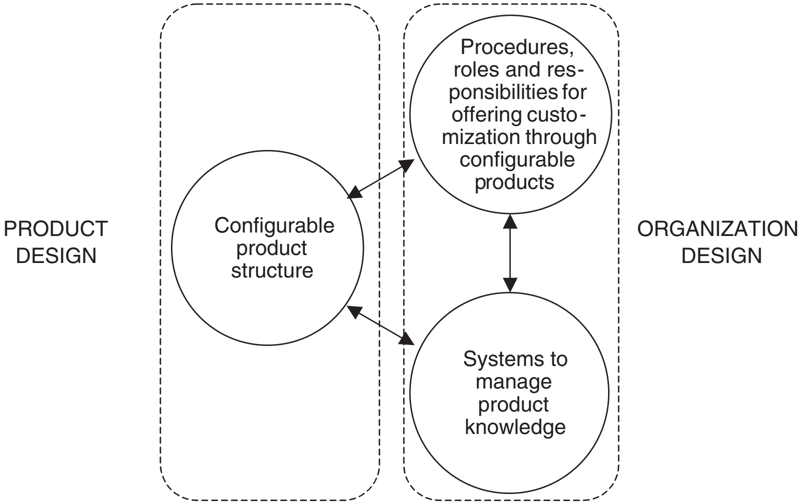


Figure 12.7: An integrative model of the technical and organizational actions needed to efficiently offer customization through product configuration

the re-engineering of the configuration process and, finally, the definition of possible inter-company interactions in the configuration process.

As shown in Figure 12.7, the adoption of the configurational approach on the part of a company is not a 'simple' technical fact, related to designing or implementing an information system. It means a great deal more, as it is a decision that involves the coordination of product design and organization design activities, including not only product knowledge management systems, but also procedures, roles, and a list of responsibilities.

Bibliography

Aamodt A. and Plaza E., 'Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches', *Artificial Intelligence Communications*, IOS Press, 1994, 7(1): 39–59.

Aldanondo, Michel, Rougé, S. and Veron, M., 'Expert Configurator for Concurrent Engineering: Camelon Software and Model', *Journal of Intelligent Manufacturing*, 2000, 11(2): 127–34.

Alford, Dave, Sackett, Peter and Nelder, Geoff, 'Mass Customisation – an Automotive Perspective', *International Journal of Production Economics*, 2000, 65(1).

Amaro, Graça, Hendry, Linda and Kingsman, Brian, 'Competitive Advantage, Customisation and a New Taxonomy for Non-make-to-stock Companies', *International Journal of Operations and Production Management*, 1999, 19(4): 349–71.

Ariano, Marco and Dagnino, Aldo, 'An Intelligent Order Entry and Dynamic Bill of Materials Systems for Manufacturing Customized Furniture', *Computers and Electronic Engineering*, 1996, 22(1): 45–60.

Attardi, Giuseppe, Cisternino, Antonio and Simi, Maria, 'Web-based Configuration Assistants: Artificial Intelligence for Engineering Design', *Analysis and Manufacturing*, 1998, 12(4): 321–31.

Barker, Virginia E. and O'Connor, Dennis E., 'Expert Systems for Configuration at Digital: XCON and Beyond', *Communications of the ACM*, 1989, 32(3): 298–318.

Berry, William L., Tallon, W. J. and Boe, W. J., 'Product Structure Analysis for the Master Scheduling of Assemble-to-order Products', *International Journal of Operations and Production Management*, 1992, 12(11): 24–41.

Bertrand, J. W. M., Zuijderwijk, M. and Hegge, H. M. H., 'Using Hierarchical Pseudo Bills of Material for Customer Order Acceptance and Optimal Material Replenishment in Assemble to Order Manufactuiring for Non-modular Products', *International Journal of Production Economics*, 2000, 66(6): 171–84.

Blecker, T., Friedrich, G., Kaluza, B., Abdelkafi, N. and Kreutler G., 'Information and management systems for product customization', New York, Springer, 2005.

Boehm, A., Mueller, H. J., Rahmer, J. and Uellner, S., A., discussion of Internet configuration systems. Proceedings of the AAAI conference, Orlando, AAAI Press, 1999.

Boynton, Andrew C. and Victor, Bart, 'Beyond Flexibility: Building and Managing the Dynamically Stable Organization', *California Management Review*, 1991, (3): 53–66.

Brown, David C., 'Defining Configuring', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1998, 12(4): 301–5.

Chao, Ping Yi and Chen, Tsung Te, 'Analysis of Assembly through Product Configuration', *Computers in Industry*, 2001, 44(2): 189–203.

Clegg, Alicia, ‘Personal Touch’, *Marketing Week*, 1999, 22(19): 45–8.
Cowan, Robin, ‘Expert Systems: Aspects of and Limitations to the Codifiability of Knowledge’, *Research Policy*, 2001, 30(9): 1355–72.
Cox, W. Michael and Alm, Richard, ‘The Right Stuff – America’s Move to Mass Customization’, Federal Reserve Bank of Dallas Annual Report, 1998.
Darr, Timothy P. and Birmingham, William P., ‘Part-selection Triptych: a Representation, Problem Properties and Problem Definition, and Problem-solving Method’, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2000, 14(1): 39–51.
Davis, Stan, *Future Perfect*, Reading, MA, 1987.
Dietrich, Glenn B., Walz, Diane B. and Wynekoop, Judy, ‘The Failure for SDT Diffusion: a Case for Mass Customization’, *IEEE Transactions on Engineering Management*, 1997, 44(4): 390–8.
Duray, Rebecca and Milligan, Glenn W., ‘Improving Customer Satisfaction through Mass Customization’. *Quality Progress*, 1999, 8: 60–6.
Duray, Rebecca, Ward, Peter T., Milligan, Glenn W. and Berry, William L., ‘Approaches to Mass Customization: Configurations and Empirical Validation’, *Journal of Operations Management*, 2000, 18(6): 605–25.
Eastwood, M. A., ‘Implementing Mass Customization’, *Computers in Industry*, 1996, 30(3): 171–4.
Elofson, Greg and Robinson, William N., ‘Creating a Custom Mass-production Channel on the Internet’, *Communications of the ACM*, 1998, 41(3): 56–62.
Erens, F. and Verhulst, K., ‘Architectures for Product Families’, *Computers in Industry*, 1997, 33(2–3): 165–78.
Feitzinger, Edward and Lee, Hau L., ‘Mass Customization at Hewlett Packard’, *Harvard Business Review*, 1997, 1: 116–21.
Feldkamp, Frank, Heinrich, Michael and Meyer-Gramann, Dieter Klaus, ‘SyDeR – System Design for Reusability’, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1998, 12(4): 373–82.
Fitzgerald, Brian, ‘Mass Customization – at a Profit’, *World Class Design to Manufacture*, 1995, 2(1): 43–46.
Fleischanderl, Gerard, Friedrich, Gerard E., Haselböck, Alois and Stumptner, Markus, ‘Configuring Large Systems using Generative Constraint Satisfaction’, *IEEE Intelligent Systems*, 1998, 13(4): 59–68.
Forza, C. and Salvador, F., ‘Managing for Variety in the Order Acquisition and Fulfilment Process: the Contribution of Product Configuration Systems’, *International Journal of Production Economics*, 2002, 76(1): 87–98.
Forza, C. and Salvador, F., ‘Product Configuration and Inter-firm Coordination: an Innovative Solution from a Small Manufacturing Enterprise’, *Computers in Industry*, 2002, 49(1), 37–46.
Franke, David W., ‘Configuration Research and Commercial Solutions’, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1998, 124): 295–300.
Freuder, Eugene C., ‘The Role of Configuration Knowledge in the Business Process’, *IEEE Intelligent Systems*, 1998, 13(4): 29–33.
Fulkerson, Bill, ‘A response to Dynamic Change in the Marketplace’, *Decision Support Systems*, 1997, 21(3): 199–214.

Galbreath, Jeremy and Rogers, Tom, 'Customer Relationship Leadership: a Leadership and Motivation Model for the Twenty-first Century Business', *TQM Magazine*, 1999, 11(3): 161–71.

Gilmore, James H. and Pine, Joseph B. II., 'The Four Faces of Mass Customization', *Harvard Business Review*, 1997, 75(1): 91–100.

Goldsmith, Ronald E., 'The Personalized Marketplace: Beyond the 4Ps', *Marketing Intelligence & Planning*, 1999, 17(4): 178–85.

Haag, Albert, 'Sales Configuration in Business Processes', *IEEE Intelligent Systems*, 1998, 13(4): 78–85.

Hansen, Claus Thorp and Riitahuta, Asko, 'Issues on the Development and Application of Computer Tools to Support Product Structuring and Configuring', *International Journal of Technology Management*, 2001, 21(3/4): 240–56.

Hart, Christopher W. L., 'Made to Order', *Marketing Management*, 1996, 5(2): 10–23.

Hart, Christopher W. L., 'Mass Customization: Conceptual Underpinnings, Opportunities and Limits', *International Journal of Service Industry Management*, 1995, 6(2): 36–45.

He, David and Kusiak Andrew, 'Performance Analysis of Modular Products', *International Journal of Production Research*, 1996, 34(1): 253–72.

Hsu, Liang H., Liu, Peia and Dawidowsky, Tim, 'A Multimedia Authoring-in-the-large Environment to Support Complex Product Documentation', *Multimedia Tools and Applications*, 1999, 8(6): 11–64.

Huffman, Cynthia and Kahn, Barbara E., 'Variety for Sale: Mass Customization or Mass Confusion?', *Journal of Retailing*, 1998, 74(4): 491–513.

Humbert, Marc, Jolly, Dominique and Thérin François, 'Building Strategy on Technological Resources and Commercial Proactiveness: the Gemplus Case', *European Management Journal*, 1997, 15(6): 658–66.

Hvam, Lars, 'A procedure for the application of product modeling'. *International Journal of Production Research*, 2001, 39(5): 873–85.

Jiao, Jianxin and Tseng, Mitchell M., 'A Methodology for Developing Product Family Architecture for Mass Customization', *Journal of Intelligent Manufacturing*, 1999, 10(1): 3–20.

Jiao, Jianxin and Tseng, Mitchell M., 'Understanding Product Family for Mass Customization by Developing Commonality Indices', *Journal of Engineering Design*, 2000, 11(3): 225–43.

Jiao, Jianxin, Tseng, Mitchell M., Duffy, Vincent G. and Lin, Fuhua, 'Product Family Modeling for Mass Customization', *Computers and Industrial Engineering*, 1998, 35(3–4): 495–8.

Jina, Jay, Bhattacharya, Arindam K. and Walton, Andrew D., 'Applying Lean Principles for High Product Variety and Low Volumes: Some Issues and Propositions', *Logistics Information Management*, 1997, 10(1): 5–13.

Juengst, Werner E. and Heinrich, Michael, 'Using Resource Balancing to Configure Modular Systems', *IEEE Intelligent Systems*, 1998, 13(4): 50–8.

Kingsman, Brian, Worden, Lee, Hendry, Linda, Mercer, Alan and Wilson, Elaine, 'Integrating Marketing and Production Planning in Make-to-order Companies', *International Journal of Production Economics*, 1993, 30–31(1–3): 53–66.

Kneppelt, Leland R., 'Product Structuring Considerations for Master Production Scheduling', *Production and Inventory Management*, 1984, 25(1): 83–99.

Kotha, Suresh, 'From Mass Production to Mass Customization: the Case of the National Industrial Bicycle Company of Japan', *European Management Journal*, 1996, 14(5): 442–50.

Kotha, Suresh, 'Mass Customization: Implementing the Emerging Paradigm for Competitive Advantage', *Strategic Management Journal*, 1995, 16(5): 21–42.

Kotler, Philip, 'From Mass Marketing to Mass Customization', *Planning Review*, 1989, 17(5): 10–13.

Lampel, Joseph and Mintzberg, Henry, 'Customizing Customization', *Sloan Management Review*, 1996, 38(1): 21–30.

Leonard-Burton, Dorothy, 'Implementation as Mutual Adaptation of Technology and Organization', *Research Policy*, 1988, 17(5): 251–67.

Leonard-Burton, Dorothy, 'Implementation Characteristics of Organizational Innovations', *Communication Research*, 1988, 15(5): 603–1.

Leonard-Burton, Dorothy, 'Implementing Structured Software Methodologies: a Case of Innovation in Process Technology', *Interfaces*, 1987, 17(3): 6–17.

Leonard-Burton, Dorothy, 'The Case for Integrative Innovation: an Expert System at Digital', *Sloan Management Review*, 1987, 29(1): 7–19.

Liechty, John, Ramaswamy, Venkatram and Cohen, Steven H., 'Choice Menus for Mass Customization: an Experimental Approach for Analyzing Customer Demand with an Application to a Web-based Information Service', *Journal of Marketing Research*, 2001, 38(2): 183–96.

L´lström, Par and Westbrook Roy, 'Implications of Mass Customization for Operations Management – an Exploratory Survey', *International Journal of Operations and Production Management*, 1999, 19(3): 262–74.

Logman, Marc, 'Marketing Mix Customization and Customizability', *Business Horizons*, 1997, 40(6): 39–44.

Lowe, Helen, Pechoucek, Michal and Bundy, Alan, 'Proof Planning for Maintainable Configuration Systems', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1998, 12(4): 345–56.

Mailharro, Daniel, 'A Classification and Constraint-based Framework for Configuration', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1998, 12(4): 383–97.

Männistö, Tomi, Peltonen, Hannu and Sulonen, Reijo, 'View to Product Configuration Knowledge Modelling and Evaluation', AAAI Press. AAAI Fall Symposium, 1996, 111–18.

Männistö, Tomi, Peltonen, Hannu, Soininen, Timo and Sulonen, Reijo, 'Multiple Abstraction Levels in Modelling Product Structures', *Data & Knowledge Engineering*, 2001, 36(1): 55–78.

Marcus, S., *Automating Knowledge Acquisition for Expert Systems*, Boston: Kluwer Academic, 1988.

Maskell, Brian H., 'Bills of Material for Engineering, Production and Costing', *Production & Inventory Management Review* with *APICS News*, 1988, 8(5): 38–44.

Mather, Hal, *Bills of Materials, Recipes and Formulations*, Atlanta: Wright Publishing Company, 1982.

McAdams, Daniel A., Stone, Robert B. and Wood, Kristin L., 'Functional Interdependence and Product Similarity based on Customer Needs', *Research in Engineering Design*, 1999, 11(1): 1–19.

McCutcheon, David M., Raturi, Amitabh S. and Meredith, Jack R., 'The Customization-responsiveness Squeeze', *Sloan Management Review*, 1994, 35(2): 89–99.

McDermott, John, 'R1: a Rule-based Configurer of Computer Systems', *Artificial Intelligence*, 1982, 19(1): 39–88.

McGuinness, Deborah L. and Wright, Jon R., 'An Industrial-strength Description Logic-based Configurator Platform', *IEEE Intelligent Systems*, 1998, 13(4): 69–77.

McGuinness, Deborah L. and Wright, Jon R., 'Conceptual Modelling for Configuration: a Description-based Approach', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1998, 12(4): 333–44.

Mittal, Sanjay and Frayman, Felix, 'Towards a Generic Model of Configuration Tasks', *International Joint Conference on Artificial Intelligence*, 1989: 1395–401.

Müller, K. and Sebastian, H. J., 'Intelligent Systems for Engineering Design and Configuration Problems', *European Journal of Operational Research*, 1997, 100(2): 315–26.

Muller-Heumann, Guenther, 'Market and Technology Shifts in the 1990s: Market Fragmentation and Mass Customization', *Journal of Marketing Management*, 1992, 8(4): 303–14.

O'Donnel, F. J., MacCallum, Ken J., Hogg, T. D. and Yu, Bei, 'Product Structuring in a Small Manufacturing Enterprise', *Computers in Industry*, 1996, 31(3): 281–92.

Oleson, John D., *Pathways to Agility: Mass Customization in Action*, New York: Wiley & Sons, 1998.

Olsen, K. A. and Saetre, P., 'Managing Product Variability by Virtual Products', *International Journal of Production Research*, 1997, 35(8): 2093–107.

Olsen, K. A., Saetre, P. and Thorstenson, A., 'A Procedure-oriented Bill of Materials', *Computers and Industrial Engineering*, 1997, 32(1): 29–45.

Peltonen, Hannu, Pitkänen, Olli and Sulonen, Reijo, 'Process-based View of Product Data Management', *Computers in Industry*, 1996, 31(3): 195–203.

Perkins, A. G., 'Product Variety: Beyond Black', *Harvard Business Review*, 1994, 72(6): 13–14.

Peters, Linda and Saidin, Hasannudin, 'IT and the Mass Customization of Services: the Challenge of Implementation', *International Journal of Information Management*, 2000, 20: 103–19.

Pine, Joseph B. II., *Mass Customization – The New Frontier in Business Competition*. Cambridge, MA: Harvard Business School Press, 1993.

Pine, Joseph B. II, Gilmore, James H. and Boynton, Andrew C., 'Making Masss Customization Work', *Harvard Business Review*, 1993, 71(5): 108–18.

Tseng, Mitchell, M. and Piller, T. Frank, 'The customer centric enterprise: advances in mass customization and personalization', Berlin: Springer Verlag, 2003.

Radder, Laetitia and Louw, Lynette, 'Mass Customization and Mass Production', *TQM Magazine*, 1999, 11(1): 35–40.

Ramaswamy, R., Ulrich, Karl, Kishi, N. and Tomikashi, M., 'Solving Parametric Design Problems Requiring Configuration Choices', *Journal of Mechanical Design*, 1993, 115(1): 20–8.

Rautenstrauch, Claus, Seelmann-Egggelbert, Ralph and Turowski, Klaus, *Moving into Mass Customization – Information Systems and Management Principles*, Springer, 2002.

Rosender, J., Armstrong, D. and Gates, J. D., *The Clickable Corporation – Successful Strategies for Capturing the Internet Advantage*, New York, NY: Free Press, 1999.

Roy, U. and Kodkani, S. S., 'Product Modeling within the Framework of the World Wide Web', *IIE Transactions*, 1999, 31(7): 667–77.

Sabin Daniel and Weigel, Rainer, 'Product Configuration Frameworks – a Survey', *IEEE Intelligent Systems*, 1998, 13(4): 42–9.

Saisse, Manoel C. P. and Wilding, Richard D., 'Short-term Strategic Management in Mass Customized Markets', *Logistics Information Management*, 1997, 10(5): 199–207.

Salvador, F., Forza, C. and Rungtusanatham, M., 'How to Mass Customize: Product Architectures, Sourcing Configurations', *Business Horizons*, 2002, 45(3): 61–9.

Salvador, F., Forza, C. and Rungtusanatham, M., 'Modularity, Product Variety, Production Volume, and Component Sourcing: Theorizing Beyond Generic Prescriptions', *Journal of Operations Management*, 2002, 20(5): 549–75.

Salvador, F. and Forza, C., 'Configuring products to address the customization-responsiveness squeeze: A survey of management issues and opportunities', *International Journal of Production Economics*, 2004, 91(3): 273–91.

Schmidt, Charles and Nourain, Janice E., 'Automating Creation and Updates Bills of Materials', *Production & Inventory Management Review* with *APICS News*, 1986, 11: 40–1.

Schonfeld, Erick, 'The Customized, Digitized, Have-it-your Way Economy', *Fortune*, 1998, 10: 114–24.

Slater, Paul J. P., 'Pconfig: a Web-based Configuration Tool for Configure-to-order Products', *Knowledge-Based Systems*, 1999, 12(5–6): 223–30.

Soberman, David A., 'It's a Whole New Ball-game', *European Management Journal*, 1999, 17(3): 290–5.

Soininen, Timo, Tiihonen, Juha, Männistö, Tomi and Sulonen, Reijo, 'Towards a General Ontology of Configuration', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 1998, 12(4): 357–72.

Spring, Martin and Dalrymple, John F., 'Product Customization and Manufacturing Strategy', *International Journal of Operations and Production Management*, 2000, 20(4): 441–67.

Srinivasan, Ramesh and Swaminathan, Jayashankar M., 'Managing Configurable Products in the Computer Industry: Planning and Coordination Issues', *Sadhana*, 1997, 22(1): 33–43.

Stonebraker, Peter W. 'Restructuring the Bill of Material for Productivity: a Strategic Evaluation of Product Configuration', *International Journal of Production Economics*, 1996, 45(1–3): 251–60.

Stumptner, Markus, 'An Overview of Knowledge-based Configuration', *AI Communications*, 1997, 10(2): 111–26.

Sviokla, John J., 'An Examination of the Impact of Expert Systems on the Firm: the Case of XCON', *MIS Quarterly*, 1990, 14(2): 127–40.

Swaminatham, Jayashankar M., 'Enabling Customization Using Standarized Operations', *California Management Review*, 2001, 43(3): 125–35.

Tatsiopoulos, I. P., 'On the Unification of Bills of Materials and Routings', *Computers in Industry*, 1996, 31(3): 293–304.

Tersine, Richard and Harvey, Michael, 'Global Customerization of Markets Has Arrived!', *European Management Journal*, 1998, 16(1): 79–90.

Tiihonen, Juha and Soininen, Timo, *Product Configurators – Information System Support for Configurable Products. Increasing Sales Productivity through the Use of Information Technology during the Sales Visit – A Survey of the European Market*, Hewson Consulting Group, 1997.

Tiihonen, Juha, Soininen, Timo, Männistö, Tomi and Sulonen, Reijo, 'State-of-the-practice in Product Configuration – a Survey of 10 cases in the Finnish Industry', in Tomyiama, T., Mäntylä, M. and Finger, S., *Knowledge Intensive CAD – Proceedings of the First IFIP WG 5.2 Workshop*, Finland, London, Chapman & Hall, 1995: 95–113.

Trappey, Amy J. C., Peng, Ting-Kou and Lin, Hsuei-Dian, 'An Object-oriented Bill of Materials System for Dynamic Product Management', *Journal of Intelligent Manufacturing*, 1996, 7(5): 365–71.

Tseng, Mitchell M. and Jiao, Jianxin, 'Case-based eVolutionary Design for Mass Customization', *Computers and Industrial Engineering*, 1997, 33(1–2): 319–23.

Tseng, Mitchell M. and Jiao, Jianxin, 'Concurrent Design for Mass Customization', *Business Process Management Journal*, 1998, 4(1): 10–24.

Tseng, Mitchell M., Jiao, Jianxin and Su, Cuan-Jun, 'Virtual Prototyping for Customized Product Development', *Integrated Manufacturing Systems*, 1998, 9(6): 334–43.

Tu, Qiang Vonderembse, Mark A., Ragu-Nathan, T. S. and Ragu-Nathan, B, 'Measuring modulairty-based manufacturing practices and their impact on mass customization capability: a customer-driven perspective', *Decision Sciences*, 2004, 35(2): 147–168.

Ulrich, Karl, Kishi, N., Ramaswamy, R. and Tomikashi M., 'Solving Parametric Design Problems Requiring Configuration Choices', *ASME Journal of Mechanical Design*, 1993, 115(1): 20–8.

Vickery, Shawnee, Dröge, Cornelia and Germain, Richard, 'The Relationship between Product Customization and Organizational Structure', *Journal of Operations Management*, 1999, 17(4): 377–91.

Ward, A. C. and Seering, W. P., 'The Performance of a Mechanical Design Compiler', *Journal of Mechanical Design*, 1993, 115(3): 341–5.

Ward, A. C. and Seering, W. P., 'Quantitative Inference in a Mechanical Design "Compiler"', *Journal of Mechanical Design*, 1993, 115(1): 29–35.

Westbrook, Roy and Williamson, Peter J., 'Mass Customization: Japan's New Frontier', *European Management Journal*, 1993, 11(1): 38–45.

Wielinga, B. and Schreiber, G., 'Configuration-design Problem Solving', *IEEE Expert*, 1997, 12(2): 49–56.

Word, Edwin G., 'Product Structures in Order Fulfilment', *Manufacturing Systems*, 1996, 12(5): 64–74.

Yu, Bei and MacCallum, J., 'Modelling of Product Configuration Design and Management by Using Product Structure Knowledge', in Tomyiama, T., Mäntylä, M. and Finger, S., *Knowledge Intensive CAD – Proceedings of the First IFIP WG 5.2 Workshop*, Finland, London, Chapman & Hall, 1995: 115–26.

Yu, Bei and MacCallum, Ken J., 'Decision Support for Configuration Management in Product Development', Proceedings of the 3rd International Conference on Computer Integrated Manufacturing, Singapore, 1995: 278–86.

Yu, Bei and Skovgaard, Jřrgen, 'A Configuration Tool to Increase Product Competitiveness', *IEEE Intelligent Systems*, 1998, 13(4): 34–41.

Index

NB: Page numbers in bold refer to figures and tables