

# Intelligent product family descriptions for business applications : production control software based upon generic bills-of-material in an assemble-to-order/make-to-stock environment

**Citation for published version (APA):**

Hegge, H. M. H. (1995). *Intelligent product family descriptions for business applications : production control software based upon generic bills-of-material in an assemble-to-order/make-to-stock environment*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Industrial Engineering and Innovation Sciences]. Technische Universiteit Eindhoven. <https://doi.org/10.6100/IR429739>

**DOI:**

[10.6100/IR429739](https://doi.org/10.6100/IR429739)

**Document status and date:**

Published: 01/01/1995

**Document Version:**

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

**INTELLIGENT  
PRODUCT FAMILY DESCRIPTIONS  
FOR  
BUSINESS APPLICATIONS**

**HERMAN M.H. HEGGE**

# INTELLIGENT PRODUCT FAMILY DESCRIPTIONS FOR BUSINESS APPLICATIONS

production control software  
based upon generic bills-of-material  
in an  
assemble-to-order/make-to-stock environment

## PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de  
Technische Universiteit Eindhoven,  
op gezag van de Rector Magnificus, prof.dr. J.H. van Lint,  
voor een commissie aangewezen door het College  
van Dekanen in het openbaar te verdedigen op  
vrijdag 13 januari 1995 om 14.00 uur

door

**HERMANUS MARINUS HENDERICUS HEGGE**

geboren te 's-Hertogenbosch

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. J.C. Wortmann

prof.dr.ir. J.W.M. Bertrand

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Hegge, Hermanus M.H.

intelligent product family descriptions for business applications: production control software based upon generic bills-of-material in an assemble-to-order/make-to-stock environment/Hermanus M.H. Hegge. - Eindhoven:

Eindhoven University of Technology.

Thesis Eindhoven. - With ref. - With summary in Dutch.

ISBN 90-386-0491-2

Subject headings: bill-of-material / product families

## **PART I Problem Definition**

<b>1 Production control in an ATO/MTS environment</b> .....	1
Introduction .....	1
Primary process .....	3
Control .....	4
Information system .....	6
Structure of this thesis .....	7
<b>2 Dealing with variants</b> .....	9
Introduction .....	9
Moving from an MTS environment to a mixed ATO/MTS environment .....	9
Primary process .....	9
Control .....	11
Information system .....	15
<b>3 Product model requirement specification for a product family</b> .....	21
Introduction .....	21
Objective of the research .....	21
Design problem .....	21
Requirement specifications for the product model .....	22
Requirement specifications for the configurator .....	23
Requirement specifications for material requirements planning .....	23
Example case study: "Desk Island" .....	24
Summary .....	31

## **PART II Product Model**

<b>4 Bill-of-material</b> .....	33
Introduction .....	33
Representation .....	34
Presentation .....	37
Path .....	41

Engineering changes . . . . .	42
Data model . . . . .	44
Points of view . . . . .	46
<b>5 Basis for the generic bill-of-material . . . . .</b>	<b>47</b>
Introduction . . . . .	47
Product family . . . . .	47
Identifying a variant . . . . .	47
Bill-of-material systems for product families . . . . .	49
Generic product . . . . .	51
Basic concept . . . . .	52
Generic primary product . . . . .	52
Generic subassembly . . . . .	53
Paths . . . . .	55
GPP coordination . . . . .	56
Inheritance . . . . .	56
Specialties associated with a generic product . . . . .	57
Forms of identification . . . . .	59
Parameters . . . . .	61
Prerequisite conditions for inheritance . . . . .	66
Data model . . . . .	68
Summary . . . . .	77
<b>6 Choice conditions . . . . .</b>	<b>79</b>
Introduction . . . . .	79
Decision trees . . . . .	79
Condition GP . . . . .	82
Complete decision tree for a GPP . . . . .	86
Evaluating the decision tree . . . . .	87
Decomposing the decision tree . . . . .	88
Decision table . . . . .	90
Multi-dimensional decision table . . . . .	93
Production rules . . . . .	94
Continuous parameter values . . . . .	94
Algorithms . . . . .	97
Engineering change . . . . .	100

Summary . . . . .	101
<b>7 Pragmatic aspects of the generic bill-of-material . . . . .</b>	<b>103</b>
Introduction . . . . .	103
Choosing a generic product . . . . .	103
Choosing generic primary products . . . . .	104
Generic product structure . . . . .	105
Parameters associated with generic subassemblies . . . . .	105
Conditions and algorithms . . . . .	106
Different points-of-view . . . . .	106
Product development point-of-view . . . . .	106
Manufacturing point-of-view . . . . .	110
Sales point-of-view . . . . .	111
Combined points-of-view . . . . .	112
Architecture . . . . .	113
Summary . . . . .	117

**PART III      Applications**

<b>8 Configurator . . . . .</b>	<b>119</b>
Introduction . . . . .	120
Configurator and bill-of-material generator . . . . .	120
Identification . . . . .	120
Configurator based upon a generic bill-of-material . . . . .	126
Starting with the top level specialty . . . . .	127
Product structure . . . . .	128
Decision path . . . . .	128
Continuous parameters . . . . .	129
Inheriting . . . . .	130
Alternative configurators based upon a generic bill-of-material . . . . .	131
The configurator and order acceptance . . . . .	134
<b>9 Material coordination . . . . .</b>	<b>139</b>
Introduction . . . . .	139
Production environment . . . . .	139
Assemble-to-order production environment . . . . .	140

1 Acquiring and manufacturing components . . . . .	141
Generic bill-of-material and planning percentages . . . . .	144
2 Production control and product demand . . . . .	147
Generic material requirements planning (GMRP) . . . . .	149
GMRP and multivariants . . . . .	151
GMRP and Line Requirements Planning (LRP) . . . . .	154
GMRP and the product demand . . . . .	155
GMRP and the configuration process . . . . .	159
3 Assembling the customer orders . . . . .	160
Uncommitted forecast . . . . .	162
Release of work order components for multivariants . . . . .	163
4 Uncertainty of demand for finished product variants . . . . .	164
Generic MRP in a mixed environment . . . . .	167
Summary . . . . .	167

## **PART IV      Evaluation**

<b>10 Comparison with other product models . . . . .</b>	<b>169</b>
Introduction . . . . .	169
Modular bill-of-material . . . . .	170
Product structure chart . . . . .	172
Logical graph . . . . .	173
Variant bill-of-material . . . . .	175
Van Veen's product model . . . . .	178
Expert system . . . . .	181
Summary . . . . .	183
<b>11 Concluding observations . . . . .</b>	<b>185</b>
Introduction . . . . .	185
Prototype system . . . . .	185
Product model . . . . .	185
GMRP . . . . .	193
Recommendations for further study . . . . .	193
Conclusion . . . . .	195
APPENDIX: GENERIC BILL-OF-MATERIAL FOR THE TEST CASE . . . . .	197



*Table of Contents*

---

<b>References</b> .....	207
<b>Notational conventions for data structure diagrams</b> .....	212
<b>Glossary</b> .....	213
<b>Summary</b> .....	215
<b>Samenvatting</b> .....	218
<b>Index</b> .....	221
<b>Curriculum vitae</b> .....	223

# 1 Production control in an ATO/MTS environment

---

## Introduction

The primary objective of the research described in this thesis is to improve the information systems that support production control in industrial environments. The research focuses particularly on the problems encountered by companies that produce and supply complex products with many possible variants. This type of production situation is often characterized as being an assemble-to-order/make-to-stock environment.

We will use the term **information system** here to mean the total of people, resources and procedures that, when combined, provide the information needed to support production control [WORT89]. An information system must first collect data in order to be able to provide information. Subsystems designed specifically for the purpose of collecting data are referred to here as the **data registration systems**. A suitable data registration system is a necessary prerequisite for the proper functioning of an information system for production control. The data registration systems are seen, thus, as important components of this type of system.

The existence of a properly functioning information system has become increasingly important in the area of production control in recent years. New features associated with information systems have contributed, in particular, to the integration of various functional areas within manufacturing companies. This integration can occur at several different levels:

- 1 at the data level. Integration here means that the data is not only available to local users, but can also be made available to every employee in the organization, if appropriate. Integration at this level also implies that the data can be kept up-to-date so that information is more timely.
- 2 at the function level. Integration at this level implies that an information system is not dedicated to a single function, but is used to support many functional areas within an organization.
- 3 between locations. Integration at this level means that an information system is distributed across more than one location so that it can support an expansive network consisting of different user sites, including sending and receiving messages within a single organization as well as between different

organizations. A system that is integrated in this way can process messages exchanged between different functional areas at different locations.

A product model is used as the basis for many information systems. The composition of a product and all of the transaction-independent product data are defined and maintained in this type of model and are subsequently made available to numerous modules of the information system. In view of this, the concept of a product model is extremely important within the context of the research described here.

Significant advances have been made in the area of production control in the last ten years. Production control is interpreted here to mean managing the flow of materials, starting from the procurement of the raw materials and parts needed in the production process and ending with the delivery of the finished products to the company department that is responsible for the external distribution to customers. Burbidge defines production control as follows: "Production control is the function of management which plans, directs and controls the material supply and processing activities in an enterprise" [BURB90]. The primary objective of production control is to realize a good delivery performance at minimal cost.

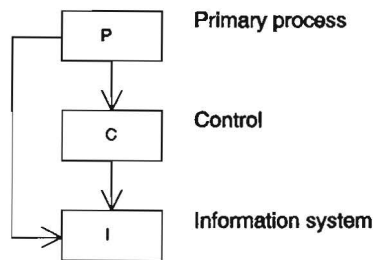


Figure 1.1: PCI Model

Choosing the best type of production control depends upon the type of production that needs to be controlled. Similarly, the choice of information system also depends upon the type of production, and additionally upon the chosen type of

---

production control. These dependencies are described in more detail in Bemelmans' PCI Model [BEME86] that is reproduced in Figure 1.1.

### **Primary process**

Numerous methods have been used to classify primary processes in different categories. The reason for defining different categories here is to simplify the task of choosing the most appropriate production control system. The classification scheme proposed by Burbidge was designed for this purpose; this is illustrated in Figure 1.2 [BURB90]. Two different distinguishing characteristics are used in Burbidge's classification scheme. The first distinguishing characteristic is used to classify primary processes into four possible categories based upon the ratio of the number of finished products to purchased products.

The categories resulting from this first distinguishing characteristic are (referring to the columns in Figure 1.2):

- the process category of primary processes represents the processes that produce a small number of finished products from a small number of raw materials;
- the imploding category of primary processes represents the processes that produce a large number of finished products from a small number of raw materials or purchased parts;
- the square category of primary processes represents the processes that produce a large number of finished products from an equal number of raw materials;
- the exploding category of primary processes represents the processes that assemble a small number of finished products from a large number of purchased parts.

The second distinguishing characteristic is used to classify primary processes into three categories based upon the repetitive nature of the production process. The resulting categories are as follows:

- jobbing production in which an individual finished product or an individual batch of finished products is produced;
- batch production in which a repetitive series of finished products is produced;

- continuous process production in which all of the finished products are produced in the form of a continuous flow.





Material Conv., Typ. Order Types	Process 	Implosive 	Square 	Explosive 
Jobbing		Jobbing foundry	Heat treat	Ship building
Batch	Beer	Glass	Electro-plate	Clothing
Continuous	Cement			Cars
Products	Bulk materials	Gen. mtl's or parts	Parts	Assemblies

Figure 1.2: Production typology (Burbidge 1990)

The type of primary process that is most relevant to the research presented here can be characterized as being an exploding process in the form of a batch or in a continuous process environment.

## Control

Burbidge provides a general description of the form of control that is most appropriate in each of the different types of production environments. He identifies the following three levels of control in this context:

- the planning level, subdivided into short term and long term planning;
- the order level, subdivided into purchase orders and production orders;
- production control at the department level.

All of these levels of control are important for the primary processes that are relevant to the research here.

The way in which production is related to the sales orders also can be used as a basis for characterizing different types of production situations. The customer order decoupling point ("CODP") [HOEK87] is the point in the flow of materials at which production control should be initiated based upon the customer orders.

---

Before this point is reached, control should be based upon the production plan. The following types of production are generally identified in this way:

- make-to-stock (MTS), whereby the CODP lies at the finished products inventory since the customer has ordered a standard product;
- assemble-to order (ATO), whereby the CODP lies at the intermediate or semi-finished product inventories since the product must be assembled based upon customer specifications;
- make-to-order (MTO), whereby the CODP lies at the stock of raw materials since the product must be manufactured to customer specifications;
- engineer-to-order (ETO), whereby the CODP is at the engineering department since the product must be specially-designed and manufactured to customer specifications.

A mixture of these types of production situations is typically found in any given manufacturing environment. We limited the scope of the research here to the MTS and ATO types of situations, or a mixture of these two types. This means that MTS products as well as ATO products may be found within a single product family.

In practice, multiple CODP's may occur instead of the expected single CODP for a given finished product. An example is when one or more components must be purchased for a customer order and the other required components may be either manufactured based upon the production plan or purchased as necessary. This type of mixed production environment will be included within the scope of this research. It is also possible that a CODP is shifted during the life-cycle of a given finished product. An example is when the demand for a newly-launched product is high enough to warrant a make-to-stock approach. This means that the CODP is at the finished product inventory. If the demand for this product diminishes in a subsequent phase of its life-cycle, however, then it may no longer be profitable to make-to-stock. An alternative would be to assemble-to-order. In this case, the CODP would shift to the stock point immediately preceding the assembly operation.

MRP-II is a control approach that is often used in an MTS or ATO situation (described in, for example, [VOLL88]). Using this approach in its original form is not a simple task, however [WEMM84].

## Information system

According to the PCI Model, choosing the proper information system depends upon the primary process and the choice of control system.

Three types of modules can be identified in an information system that supports production control [BERT90]:

- state-independent transaction processing (SITP) modules  
The "state" refers to the current status of the orders and materials in the primary process. SITP modules maintain descriptions that are independent of any specific transaction, such as the product description, the description of the process used to manufacture the product and a description of the resources required to carry out the process.
- state-dependent transaction processing (SDTP) modules  
The SDTP modules collect and record the data associated with the flow of materials and the orders. Examples of SDTP modules include Purchase Registration, Production Registration, Sales Registration and Inventory Registration.
- decision support (DS) modules  
The DS modules support the decisions that need to be made in connection with production control. Examples of DS modules include Master Production Planning, Material Requirements Planning, Capacity Requirements Planning, Order Acceptance and Work Order Release.

The research presented here focuses primarily on the Product Description Module. The bill-of-material provides a standard product description in an ATO/MTS situation. A bill-of-material also defines how a product is constructed in terms of components. The actual way of presenting a bill-of-material depends on factors such as the point-of-view from which a product is seen. An engineer typically takes a different point-of-view than a production planner. The various aspects of a bill-of-material are explained in more detail in Chapter 4.

---

A large number of the modules that comprise an information system make use of the bill-of-material. To start with, the Material Requirements Planning Module uses the bill-of-material intensively. The bill-of-material is used in this case to explode the material requirements of the finished products into the material requirements of the corresponding components. The work orders for the various production departments then can be generated based upon this information.

It is obvious that the SDTP modules necessarily rely upon the information in the bill-of-material to a great extent since they keep track of the flow of materials.

In addition to the production control functions, a variety of other disciplines within a company typically make use of the bill-of-material. An Administration and Control Department may use the bill-of-material as the basis for calculating cost prices for finished products as well as intermediate products. The cost of materials in the cost price can be determined based upon the bill-of-material. Similarly, a Research & Development Department may use the bill-of-material to define a product in terms of its basic functions. A Service Department can use the bill-of-material to find out which components are incorporated in the finished products that have already been delivered to customers. Furthermore, the bill-of-material can be used to determine the necessary contents of specific service kits.

### **Structure of this thesis**

The average number of product variants that need to be supported in an ATO/MTS environment has been increasing drastically in recent years. This increase has caused a variety of problems with respect to the primary processes as well as for the control systems and the information systems. The most serious problem for the information systems is maintaining the product description data for all of the possible product variants. This problem of dealing with a large number of variants is covered in more detail in the next chapter.

The structure of this thesis is explained at the end of this chapter. The thesis has been divided into four separate parts. Part I describes the environment for which the development of a product model is proposed. In addition, a requirement specification is derived for such a product model. The actual product model is subsequently developed in Part II. Two applications are then discussed in Part III



that are relevant with respect to the chosen production environment. Finally, in Part IV, the proposed product model is evaluated.

Part I consists of Chapters 1 through 3. A definition of the scope of the production environment in which the product model is to be used is presented in Chapter 1. Chapter 2 describes the issues that arise when the number of finished product variants is increased. The requirement specification for the product model is provided in Chapter 3.

### **Introduction**

The problems of dealing with a large variety of product variants are described in this chapter. Starting with a situation in which only standard products are produced and subsequently supplied to customers from stock, the option of assembling a portion of the products to customer orders is then considered. This results in a mixed assemble-to-order/make-to-stock (ATO/MTS) environment. One of the major problems is the subsequent maintenance of product descriptions.

### **Moving from an MTS environment to a mixed ATO/MTS environment**

The consequences of converting from one type of production environment to another can be identified in three different areas:

- 1 the consequences for the primary process;
- 2 the consequences for the control system;
- 3 the consequences for the information system.

### **Primary process**

The most important consequence for the primary process is that the assembly of the product is driven by customer orders as well as stock replenishment orders. Such customer orders are typically smaller than replenishment orders with respect to the lot size. Furthermore, the total number of finished products to be assembled will increase significantly. After all, this was the reason for converting to an ATO environment in the first place.

The assembly function must be able to deal with producing a large variety of products in quick succession. In other words, the assembly lead time must be short in order to ensure short delivery times. Other lead times are also important in addition to the assembly lead time in order to realize a short delivery time. The delivery time is generally equal to the sum of the lead times of the following processes:

- 1 customer order acceptance;
- 2 production planning for the customer order;
- 3 assembly of the customer order;
- 4 delivery of the customer order.

The largest portion of the lead time typically consists of wait time. Attention needs to be focused, in the first place, on reducing the wait times in order to shorten the total lead times. The following options for reducing the total wait time have been identified by Bertrand and Wijngaard [BERT89]:

- a: reduce the number of sequential congestion points;
- b: lower the average required processing time;
- c: reduce the capacity loading percentage;
- d: decrease the variance in the processing time.

In order to reduce the capacity loading percentage for product assembly, for example, excessive capacity could be employed or provisions made to be able to quickly adjust the capacity whenever necessary to deal with any extreme fluctuations in the customer demand. Such adjustments to processing capacity could be realized, for example, by asking employees to work overtime, creating additional shifts or calling up part-time workers. Another method of ensuring sufficient capacity to be able to deal with unanticipated peaks in customer demand is to assemble more MTS products during periods in which the customer demand is relatively low. In this way the inventory of MTS products can be used as a buffer for the production capacity. A third possibility is to lengthen or shorten the delivery times for the customer orders.

An additional method of reducing the average processing time for delivery is to choose a product assembly location that is closer to the customer plants.

A sufficient supply of components should be available, of course, to assemble the customer orders and the replenishment orders. Generally, this can be achieved in two ways. One way is to manufacture or purchase a sufficient number of components to maintain adequate stock levels. Another way is to manufacture or purchase the components to customer order. This implies that the sum of the delivery time for the components plus the remaining lead time after assembly of the components will be less than the available lead time for assembly of the finished product. This option is more attractive than the first alternative since no component inventories are required. In this case it is necessary to maintain a good network of suppliers, internally as well as external to the company. It is important to note that this second alternative cannot be considered a true ATO environment since the CODP lies with the suppliers rather than at the component inventory.

---

The product assembly function in an ATO/MTS environment may turn out to be more complex than in an MTS situation due to the presence of more orders on the shop floor, a larger quantity and variety of components on the shop floor and a larger number of different finished products to be assembled. There is a larger variety of finished products as well as components.

### **Control**

As mentioned in the previous chapter, production control in an MTS or ATO environment is often based upon the principles of MRP-II in view of the current views on production control. The framework associated with this type of approach is illustrated in Figure 2.1 and described in [VOLL88].

The most important objective of production control in situations with a large variety of finished products is to ensure that the required components are available when they are needed. For this reason, it is of critical importance to document and maintain the details of how the finished products are constructed.

The description of a product in an MTS or ATO environment is normally based upon the traditional bill-of-material. In some instances the products are grouped into product families. Such product families are then defined and documented as pseudo-products in a bill-of-material system. A Product X can then be linked to a product family via a bill-of-material relationship. The planning percentage can also be stored and maintained via this bill-of-material relationship. This percentage indicates how much of the total finished product demand for this product family is for Product X. This type of bill-of-material for a product family is referred to as a percentage bill-of-material [VEEN91].

The finished products in an MTS environment are typically the MPS (Master Production Scheduling) items. The forecasted demand for these items is determined based upon the product family and the percentage bill-of-material. In connection with this, it is assumed that the distribution of the demand for the product family across the related finished products will remain constant in time. This greatly simplifies the forecasting effort required by the user, provided that the percentages in the percentage bill-of-material generally remain unchanged. This is certainly the case when one is not able to base his projections on a time-series of historical data.

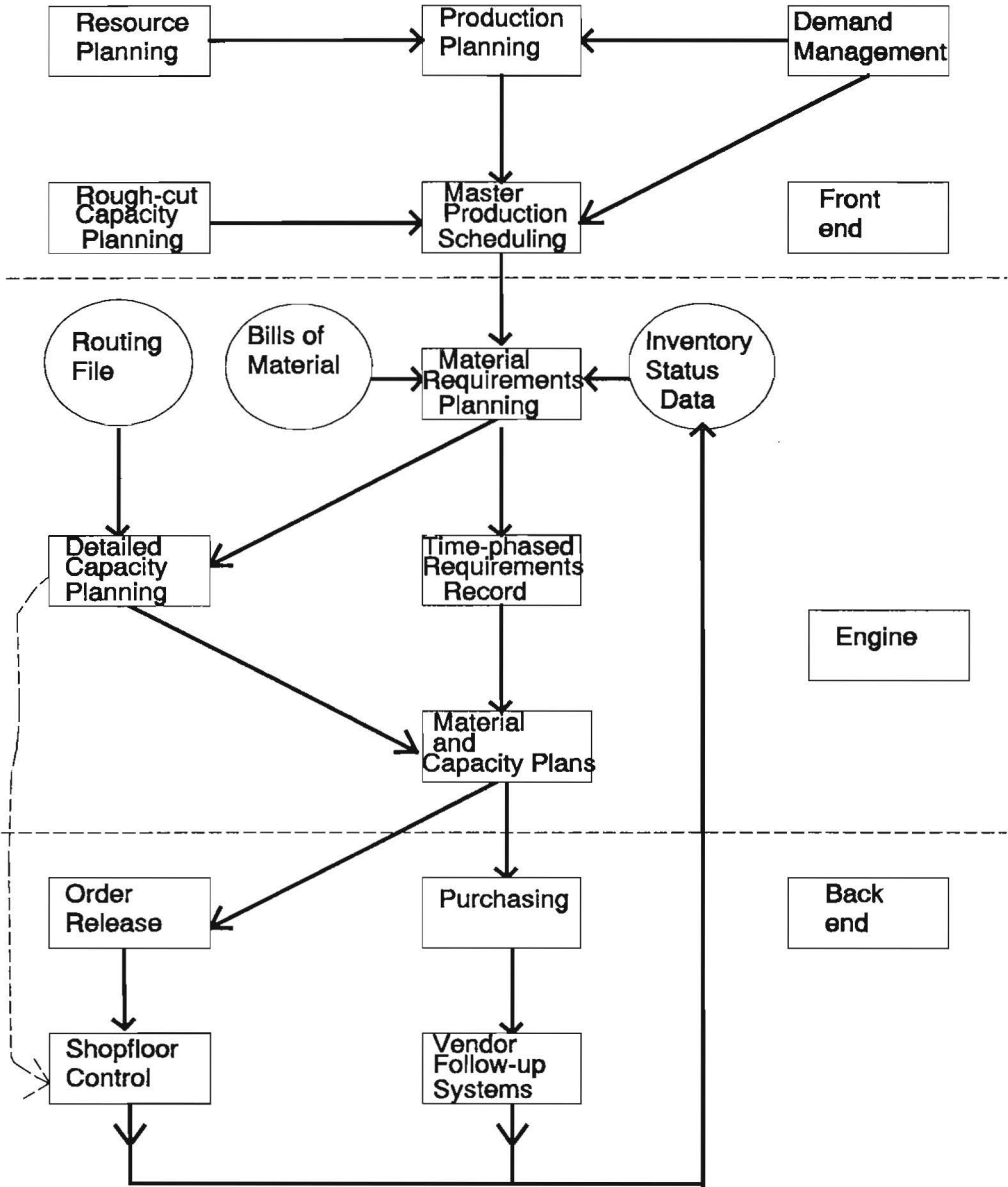


Figure 2.1: Basic concept of MRP-II [VOLL88]

---

This is a common occurrence in practical situations, especially in situations such as when an old product family is replaced by two or more new product families.

When there is a significant increase in the number of product variants and an MTS production environment is converted to an ATO environment, it becomes difficult to assign an identification code and to define a separate bill-of-material for every possible variant. The normal solution to this problem is to make use of features and options to identify a specific variant within a product family. A **feature** can be viewed as a product family characteristic for finished products. Each feature (e.g., the color of the product) is associated with a set of mutually exclusive options (for example, a choice from the colors red, gray or blue). An **option** is, thus, a value assigned to a characteristic that represents a property of a product that is normally relevant for a customer.

In this case, the MPS items are typically found at the level of major components or important subassemblies [ORLI72]. Forecasting the demand for the finished products is often based upon the forecasted demand for the various options. Translating the option forecasts to MPS item forecasts is accomplished using so-called modular bills-of-materials. The MPS items are grouped in planning modules in a modular bill-of-material. A **planning module**, therefore, becomes either a set of products needed to manufacture a finished product with a certain option or combination of options within a product family or, otherwise, a set of products that is used in every finished product in the family [VEEN91].

A forecast for a planning module is generated in MRP-II based upon the forecast for the whole product family, multiplied by the planning percentage associated with the planning module. This percentage is calculated by multiplying all of the percentages assigned to the respective options, together.

Van Veen has pointed out which assumptions are inherent in the use of a modular bill-of-material [VEEN91]. These assumptions are as follows:

- 1 There is a one-to-one relationship between a planning module and an option. The bill-of-material associated with a planning module contains all of the components needed to manufacture the product with that option.

- 2 There are no dependencies between the options associated with different features. This implies that any option can be chosen for a given feature without affecting the choices of options for other features when an order is place. In other words, a orthogonality exists between the features.
- 3 No common items are found at the lower levels of the bills-of-materials for the options belonging to a given feature.
- 4 An assembly bill-of-material can be defined for a variant of a finished product even when a modular bill-of-material is used.

Van Veen has shown that these assumptions often do not hold true when there are a large number of variants within a product family. As the number of variants increases, the number of options will normally increase and the likelihood of inter-dependencies will increase. Furthermore, it will become increasingly difficult to define bills-of-materials for product assembly based upon the modular bill-of-material in this case. Besides, there will be a tendency to purchase or manufacture as many components as possible to customer order. The CODP will move upstream in the flow of materials. To realize this, bills-of-materials with multiple levels will be needed by manufacturing. This means that bills-of-materials will be needed for the manufacture of components as well as for assembling the product.

The following functions in the MRP-II framework become the focal point for control in an environment with a large number of product variants:

- Master Production Scheduling (MPS);
- Material Requirements Planning (MRP);
- Demand Management (including Customer Order Acceptance);
- Final Assembly Scheduling (FAS) (the Material and Capacity Plan of the final assembly);
- Order Release;
- Purchasing.

The product descriptions play an important role in all of these functions. With respect to the MPS function, the planned bills-of-materials are used as the basis for translating the product family forecasts to finished product forecasts or planning module forecasts. In the case of MRP, however, the bill-of-material is used to

---

translate the forecasts for finished products and/or planning modules to forecasts for the related components and raw materials. The bill-of-material is used in the FAS function as the basis for constructing the assemblies and subassemblies. The Customer Order Acceptance function makes use of the bill-of-material to calculate the expected delivery date; the bill-of-material is needed to determine the quantities of components required to assemble a given customer order and whether a sufficient quantity of components will be available in stock. With respect to the Work Order Release function, the bill-of-material is used to allocate the necessary components.

### **Information system**

The consequences for the information system are extremely severe when the primary process is required to adapt to a large variety of finished products. These consequences are described here using the modular structure introduced in Chapter 1.

#### *1 state-independent transaction processing (SITP) modules*

As previously seen, these modules describe the product, the process and the resources needed to carry out the process. An increase in product variety will primarily affect the descriptions of the product and the process. It becomes impractical to document the specific descriptions for every possible product variant. Besides, only a limited number of the theoretically possible product variants will actually be manufactured. This means that it would be a waste of time to prepare descriptions for all of the possible variants. The best approach is to document only the descriptions for the product variants that will actually be manufactured. So-called **generative systems** have been developed to automatically provide the descriptions for product variants as required. This type of system can be used to generate a specific bill-of-material and a specific routing. Van Veen describes a number of systems that can be used to generate specific bills-of-materials [VEEN91].

In addition to the problem of accommodating a large number of variants at the finished product level, this problem may also occur at other levels within the flow



of materials. This implies that it may also be better to generate descriptions at these other levels only when this documentation is actually required.

### *2 state-dependent transaction processing (SDTP) modules*

These modules collect and record the data associated with the flows of materials and orders. The number of transactions on the shop floor increases as the number of product variants increases; the number of transactions is directly related to the number of customer orders, particularly the assembly orders. It is not possible to determine *a priori* whether all of these transactions need to be recorded. Several system approaches are based upon an extremely limited registration of data in relation to the transaction volume (e.g., KANBAN). In the event that every transaction needs to be recorded in the system, then a product identification will always be required. Use of a unique product code is the simplest approach. This means that a suitable product code must be assigned to each product, either manually or automatically. Such product codes may not be assigned randomly since any given product variant should always be referenced by the same product code. A number of alternative product identification schemes are discussed in Chapter 5.

Of course, the SDTP modules should make use of the product descriptions of the SITP modules.

### *3 decision support (DS) modules*

An increase in the number of product variants affects a large number of the DS modules. In particular, the modules that support the product-oriented functions within the MRP-II framework will need to be modified. As previously indicated, these functions are Master Production Scheduling, Material Requirements Planning, Final Assembly Scheduling, Customer Order Acceptance, Order Release and Purchasing.

The MPS results in a plan that is approved by Sales as well as Production. This plan needs to be formulated in aggregate terms in an ATO environment due to the large number of product variants. The creation of "product families" is the straightforward way to accomplish this. Since a product family is often defined at a level that is too high for this purpose, it becomes necessary to take the specific

---

properties of individual products into account. The distribution of such properties within a product family can be expressed in terms of planning percentages. A planning percentage may refer to a single property or a combination of properties. The definition of the properties of a product for this purpose must, of course, be from the customer's point-of-view rather than from a technical point-of-view.

In most cases, certain combinations of properties may not be permitted. This may be due to the fact that certain combinations of properties are physically infeasible. Other combinations may not be acceptable due to marketing restrictions. In any case, it must be possible to document such conditions and restrictions. The presence of conditions and restrictions will normally affect how the planning percentages are defined since it may be necessary to differentiate product sub-families based upon specific combinations of properties.

In general, conditions and restrictions can be formulated in two ways: in a positive way or in a negative way. A negative formulation is one in which the invalid combinations are specified, while a positive formulation states the valid combinations. Our preference here is to express the conditions in a positive way in terms of the valid combinations since this then provides a convenient basis for assigning aggregate planning percentages.

Besides the need to aggregate finished products, it is also necessary to make provisions for aggregating products at lower levels in the bill-of-material. This means that it is desirable to generate an MPS at the subassembly or component level as well as at the finished product level [BERT92]. When generating an MPS in a mixed ATO/MTS environment, it should be possible to focus on individual products within a product family, at the finished product level as well as at lower levels in the bill-of-material, in addition to the possibility of planning at the product family level.

Following from this, the product description needed in connection with generating an MPS should allow for the following possibilities;

1. the definition of product families at any hierarchical level in the product structure;
2. the definition of properties associated with a product family;
3. the specification of valid combinations of properties;

4. the specification of planning percentages per property or combination of properties;
5. the definition of individual products within a single product family.

The large variety of finished products has a significant effect upon the coordination of materials at the component and raw materials level as well as at the finished product level. This is logical since the product variants are the direct result of using alternative components at lower levels in the product structure. A greater variety of products implies that the average demand for a single product will be less. The uncertainty of the demand at the product level will increase. MRP becomes more difficult in situations of uncertain demand [BERT90]. In general, however, the uncertainty of the demand for a product family will be less than uncertainty associated with each of the individual products. This makes it more attractive to perform the requirement planning at the product family level rather than at the individual product level. This is certainly the case for product families with many common components. This means that it is important to be able to formulate plans at the product family level in an ATO environment. On the other hand, planning will need to be done more at the individual product level in an MTS environment. It is particularly convenient when the bill-of-material for each product belonging to a given family in an MTS environment can be generated based upon the product family description. In other words, the aggregation of products is also useful for the coordination of materials.

The Final Assembly Schedule (FAS) is used to drive the final stage of production. The accepted customer orders provide the basis for generating this schedule in an ATO environment. In an MTS environment, however, either the planned production of replenishment orders for the finished product inventory or the MPS is used as the basis for FAS. The materials and the resource capacities needed in the assembly process are specified in the FAS. The assembly process may involve the production of one or more subassemblies and a final assembly. For this reason, a product description may be required at several levels during the assembly process; this is not a requirement in the case of a traditional, modular bill-of-material.

A number of data elements need to be recorded when a customer order is accepted. To start with, the type of product ordered by the customer must be recorded in terms that the customer is able to understand. These are referred to as the sales

---

order parameters. The sales order parameters may not always be identical to the technical parameters that are used by manufacturing. A facility is, thus, needed to translate sales order parameters to the required technical parameters.

A second data element to be recorded is the sales price. It is often necessary to calculate the cost price of a product before the sales price can be determined.

Other data elements to be recorded include the quantity and the delivery date; this requires a link between the customer order acceptance function and the production scheduling function. This link may be weak or strong. A weak link exists when the customer order decoupling point (CODP) is used to verify the availability of materials. The CODP is at the finished product inventory in the case of MTS products. For ATO products, however, the CODP is located at the inventory of components that immediately precedes the assembly function. In the case of a strong link, the availability of sufficient materials can be verified at each stock point in the flow of materials.

The conclusion can be drawn that the customer order acceptance function requires the formulation of a product description in terms of sales order parameters that, subsequently, need to be translated into the corresponding technical parameters.

The large number of product variants in an ATO environment also has an effect upon the release of work orders. The work order for the required assembly tasks must be initiated by the customer order in an ATO environment. The demand for a large number of identical products in a given period will be significantly lower in this case than in a situation with standard products. This means that if the batch size is the same in both cases, a batch of a given product will be produced less frequently in an ATO environment. This can easily lead to an excessive amount of work-in-progress and long lead times. This can be compensated somewhat by reducing the set-up times and/or manufacturing several variants of products within a family simultaneously. Manufacturing variants simultaneously suggests that a single work order should be released, specifying a number of variants belonging to a single family. In other words, a type of aggregation is also needed here.

The purchasing function has undergone a number of drastic changes in the past decade, particularly in ATO environments. A clear trend is to limit the number of

suppliers to only a few selected vendors and to develop a close working relationship with these vendors. Increasingly more information is exchanged that deals with how the aggregate demand for materials will fluctuate over a longer period of time, in addition to the normal information related to specific purchase orders (such as the picking slip and invoice [KORN92]). Information that is useful at the planning level is exchanged as well as the information needed at the operational level. This means that information is communicated to the supplier about material planning for product families. The supplier can then use this information for his medium-term capacity planning. Similar to the situation with work orders, the purchase orders will need to incorporate multiple product variants belonging to a given product family [KREU94].

In conclusion, it can be stated that it is desirable to use aggregate information in an ATO environment with many product variants, particularly with respect to the MPS, material coordination and work order release functions. It must be possible to define this aggregate information and include this in the design of a product model. The requirement specification for such a product model is developed in the next chapter.

## 3 Product model requirement specification for a product family

---

### Introduction

To start with, the objective of the research is formulated in this chapter. A central design problem is derived from this. This design problem is subsequently translated into three separate design issues that deal with specific aspects of the product model. A requirement specification is then developed for each of these design aspects.

### Objective of the research

A number of problems that occur with respect to production control in an ATO/MTS environment were discussed in the last two chapters. This has led to formulating the objective of the research presented here as follows:

*The objective of the research is to improve the information systems that support production control in industrial environments. The research focuses particularly on the problems encountered by companies that produce and supply complex products with many possible variants. This type of production situation is often characterized as being an assemble-to-order/make-to-stock environment.*

### Design problem

Production control can be characterized as being material-oriented in an environment as described above. This means that the primary control focus will be on coordinating the different flows of materials. The allocation of resource capacity is of secondary importance. The availability of a proper product description becomes essential for the material planning function. This description is needed for carrying out the manufacturing operations as well as for controlling production. Due to large number of product variants, the product model should be based upon the description of a product family. The product family description can then be used to generate a description for a specific product. This leads to the following formulation of the design problem to be addressed here:

*Design a product model for a product family for use in an assemble-to-order/make-to-stock production environment. Using this model, it should be possible to configure a specific product so that it can be manufactured and the material requirements can be planned appropriately.*

In other words, three separate design issues need to be resolved:

- 1 designing a product model for a product family;
- 2 designing a configurator (including a bill-of-material generator);
- 3 material requirements planning based upon the product model.

The most important results of this research will be incorporated in a prototype in which these three design issues are addressed and resolved.

### **Requirement specifications for the product model**

A number of requirements for the product model have already been identified in the previous chapter. These requirements are summarized below. This product model must support:

1. the definition of product families at all levels of the hierarchical product structure;
2. the specification of product properties for each product family;
3. the specification of planning percentages for each property or combination of properties;
4. the definition of individual products within a single product family;
5. a specification of which combinations of product properties are valid;
6. the translation of one set of product properties into a different set of product properties.

A case study example called "Desk Island" has been developed to test the results of this research of [MOUL89]. This case is described in detail at the end of this chapter. All of the requirements that have been imposed on the product model are included in this case study. A total of 300,000 different finished product variants can be configured within the "Desk Island" product family.

### **Requirement specifications for the configurator**

1. Support the identification and formulation of a feasible, valid product variant based upon the product model.
2. Generate a bill-of-material based upon the variant specification and the product model.

### **Requirement specifications for material requirements planning**

In general, the material requirements planning approach to be developed should be applicable to an MTS, ATO or mixed ATO/MTS environment. This implies that the following capabilities are required (refer to Chapter 2 for a further explanation):

1. generation of a material requirements plan based upon the definitions of finished products, components and product families;
2. generation of a material requirements plan, if necessary, for a product variant within a product family;
3. ability to use planning percentages per property or combination of properties;
4. ability to generate plans based upon for aggregate information;
5. ability to release individual work orders based upon aggregate work orders;

When planning is carried out at an aggregate level, recommendations for releasing aggregate work orders will also be generated. Such aggregate work orders will need to be split up into individual work orders for the different product variants found in the aggregate order.

6. ability to drive the final stage(s) of production of a finished product based upon customer orders as well as stock replenishment orders.



**Example case study: "Desk Island"**

(borrowed from [MOUL89])

As previously mentioned above, this example case situation has been developed specially for the purpose of the research presented here. All of the requirements listed in the requirement specifications for the product model have been included in the requirements of this example case.

A wide range of office furniture is offered by the ..... Company. The Desk Island product family is just one of their many products. Each Desk Island consists of at least one or as many as four desk units. The desks are manufactured from wood in four standard sizes (lxwxh):

(180x80x90)

(180x90x90)

(200x80x90)

(200x90x90).

The desks included in a single Desk Island must all be the same size. Each desk can be supplied with one or two drawer units. If only one drawer unit is ordered, then it can be mounted under either the left side or the right side of the desktop.

The drawer unit is available in two models:

- with one pen drawer and two standard drawers; or
- with one pen drawer and one file drawer for hanging folders.

(Refer to Figures 3.1 and 3.2 for illustrations of the desk and drawer units.)

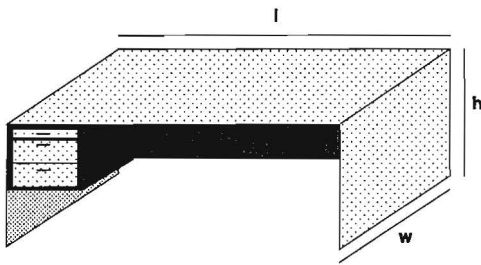
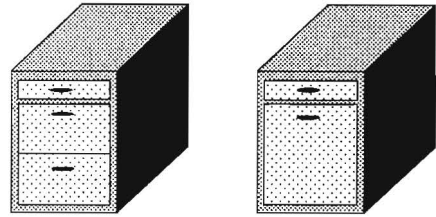


Figure 3.1: Desk



Drawer unit:  
 - one pen drawer  
 - two standard drawers

Drawer unit:  
 - one file drawer  
 - one pen drawer

Figure 3.2: Drawer units

The desks that are combined to form a Desk Island are connected together using connecting segments. Three different types of connecting segments can be used:

- quarter-round: - for connecting two desks at a 90° angle;
- triangle: - for connecting two or three desks at a 120° angle;
- square: - for connecting two, three or four desks at a 90° angle.

(Refer to Figure 3.3 for illustrations of the different connecting segments.)

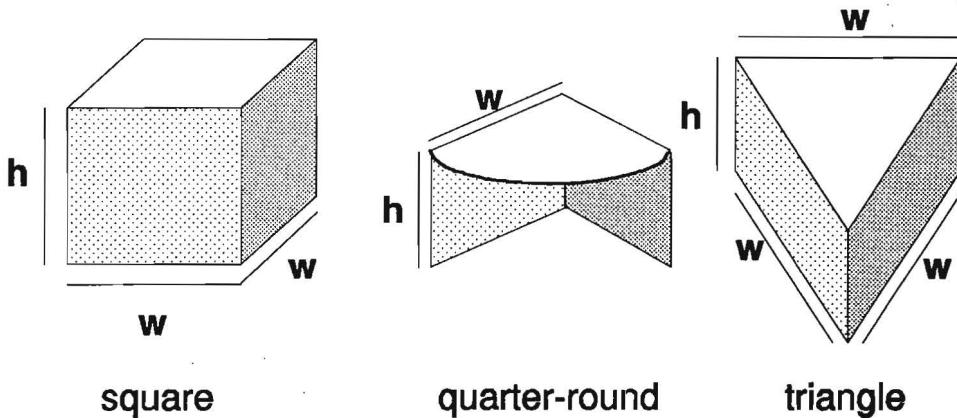


Figure 3.3: Connecting segments

Table extensions can be added to the desks if desired. This extension can be attached to either the left side or the right side of a desk. Two variants of the table extension, mirror images of each other, can be provided (see Figure 3.4).

A table extension can be supplied with or without a drawer unit. When a table extension is added to a desk, the desk may only have one drawer unit (see Figure 3.5 for an illustration of a desk with table extension). A drawer unit can then be installed under the side opposite from the table extension. The feasible desk configurations are indicated by an asterisk ("\*") in the table below:

		table extension quantity and position		
		one, left	one, right	none
drawer unit quantity and position	one, left	--	*	*
	one, right	*	--	*
	two	--	--	*

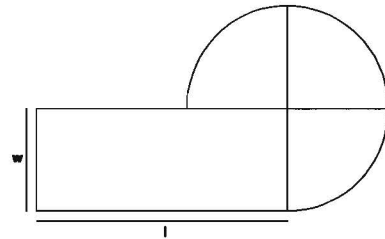
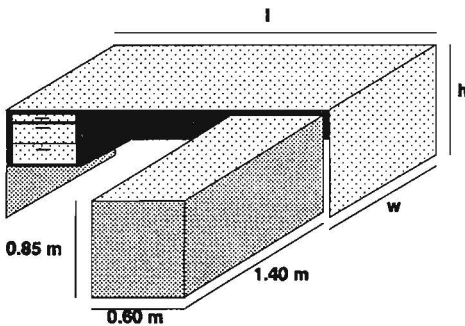


Figure 3.6: Desk in P-configuration

Figure 3.5: Desk with table extension

When multiple desks are joined via connecting segments in a Desk Island, all of the desks are supplied either with or without table extensions. Furthermore, the table extensions in a Desk Island with multiple desks cannot be attached arbitrarily. The valid combinations are indicated in the table below.

number of desks	table extension position for desk number			
	1	2	3	4
2	L	R	-	-
	L	L	-	-
	R	R	-	-
3	L	L	L	-
	R	R	R	-
4	L	L	L	L

(L = table extension on the left; R = table extension on the right)

A desk can be supplied in a so-called P-configuration by attaching three quarter round connecting segments at one of the ends. Only desks with a length of 2.00 meters can be provided in a P-configuration. In fact, all desks with a length of 2.00 meters are provided in a P-configuration. The P-configuration is only valid for single, unattached desks.

(Refer to Figure 3.6 for an illustration of a desk in a P-configuration.)

**MODIFICATIONS:**

**I: Different materials:**

Desk Islands can be manufactured from laminated composition board instead of solid wood if desired. These so-called laminated Desk Island configurations can be supplied in various colors: light gray, gray, black and brown. All of the parts of a given Desk Island are manufactured from the same material and supplied in the same color.

**II: Different colors:**

The connecting segments can also be supplied in red or blue. All of the parts of a Desk Island have the same color except for the possibility of having

connecting segments with a different color. The valid color combinations are indicated by an asterisk ("\*") in the table below.

color of connecting segments	color of desk and other parts			
	light gray	gray	black	brown
light gray	*	--	*	--
gray	--	*	*	--
black	*	*	*	--
brown	--	--	--	*
red	--	*	*	--
blue	--	*	*	--

When a quarter round segment is used in a P-configuration, then this part always has the same color as the desk.

### III: Desk chairs:

Matching desk chairs in red, gray, blue, brown or green can be supplied with the desks. One chair can be supplied for each desk. All of the desk chairs are swivel chairs on wheels. Three non-swivelling chairs without wheels can be provided with each desk in a P-configuration.

The following color combinations are valid for the laminated desk models with desk chairs:

chair	desk			
	light gray	gray	black	brown
red	*	*	--	--
gray	--	--	*	*
blue	*	*	--	--
brown	--	--	*	*
green	*	*	--	*

Each chair is constructed as follows:

A chair consists of a seat, a back, a underframe and optional armrests (an add-on option) and can be supplied with or without swivel and with or without wheels. A chair with wheels is always a swivel chair, however. The seat and the back are constructed using seat and back frames with decorated upholstery. The seat and back always have the same color. The underframe consists of a stand (two variants, depending on whether this is a swivel chair) and optional wheels (five).

The valid chair configurations are presented in the table below:

wheels	upholstery color					
	swivel	red	gray	blue	brown	green
yes	yes	*	*	*	*	*
	no	--	--	--	--	--
no	yes	*	*	--	*	*
	no	--	*	--	*	*

IV: Variable width:

The desks can be supplied is a variety of widths from 80 cm. to 100 cm. in steps of 0.5 cm..

V: Variable height:

The height of the desks is variable between 80 cm. and 100 cm. in steps of 0.5 cm.. The height of an optional table extension is always based upon the height of the desk, whereby the difference in height between the desk and the table extension is always 5 centimeters.

**Components of the various parts of a Desk Island:**Desk:

- desktop, lxw
- two sides, wxh
- connecting piece, lx60
- one or two drawer units

Drawer unit:

- back, 60x45
- two sides, 70x60
- bottom and top, 70x45
- front, 60x45
- pen drawer
- two standard drawers or a file drawer for hanging folders
- two or three pairs of drawer rails
- lock

Table extension:

- table top, 140x60
- back, 140x85
- side, 140x85
- one or no drawer unit

Quarter round connecting segment:

- quarter round top, b
- two sides, wxh

Triangle connecting segment:

- triangular top, b
- three side pieces, wxh

Square connecting segment:

- square top, wxw
- four side pieces, wxh

All of the flat components are produced from standard panels measuring 200x200 cm..

**Summary**

The objective of the research has been explained in this chapter. From this, the central design problem has been identified in terms of the need for a product model for a product family. An example case situation, "Desk Island," has been defined as a basis for testing the product model. The product model to be developed must be able to support assemble-to-order/make-to-stock production environment. In addition to the design of a product model, two other aspects are important: the design of a configurator and the design of a suitable material requirements planning approach based upon the product model.

Part II of this book focuses on the development of the product model. The traditional product model used in information systems for production control is the bill-of-material. A number of basic terms used in connection with the bill-of-material are explained in Chapter 4. The generic bill-of-material developed in connection with this research is described in Chapter 5. Chapter 6 concentrates solely on the registration of the valid combinations of product properties. Finally, Chapter 7 describes how a product family can be modelled based upon a generic bill-of-material.



### Introduction

This chapter starts with a formal definition of the term "bill-of-material." A number of aspects of a bill-of-material are then discussed such as the manner of representation, the presentation, the meaning of a "path" and making engineering changes. This leads to the formulation of a data model for a bill-of-material. Finally, we show that a product can be represented in more than one way.

### Product structure

The available knowledge of products and processes needs to be recorded and documented in every industrial company. In other words, a company should document the information about the products it makes, or can make, and how this is accomplished.

The normal way of documenting this product knowledge is in the form of a **product structure**. Our definition of a product structure is:

a model of a product that represents the way in which the whole product is constructed by a company from a specific point-of-view.

This means, among other things, that the construction of a product is not necessarily restricted to only the activities that are carried out within the company, itself. External activities may be included which are performed at the customer site or at a supplier's plant. A typical example of an external activity is the assembly and installation of equipment at the customer's plant. An example of an activity at the supplier's plant might be the construction of a subassembly that purchased as a product component but is manufactured completely to the specifications of the buying organization.

A **finished product** is a product that a company can sell to its customers. This definition implies that an intermediate product can also be a finished product when it can be sold as such to a customer.

The **primary products** are found at the lowest level of the visible product structure for a company. These primary products are usually the raw materials and the purchased parts that a company buys from suppliers. Nevertheless, it is possible that the purchased parts also have their own product structure. At least one primary product can be identified in a product structure. The finished products and the primary products provide the links between a company and its external environment with respect to the flow of materials (see Figure 4.1).

The bill-of-material is one of the most popular ways of representing a product structure.



Figure 4.1: Flow of materials relationships

A finished product may be constructed in stages. In this case, a **subassembly** of a finished product can be identified at each stage. Such subassemblies may also be referred to as semi-finished products or intermediate products.

### Representation

A bill-of-material is typically defined using two entity types, namely, product entity type and bill-of-material relationship entity type.

Data concerning the primary products, subassemblies and finished products are associated with each **product** entity. A product is normally identified by a **product code**. The required attributes of a product are a product description and a unit of measure in which the product quantity can be expressed.

---

Each **bill-of-material relationship** entity defines a relationship between a given product ("Product P") and one of the products used directly to construct this product. Product P in this case is referred to as the **parent** of the bill-of-material relationship. Each of the products used, directly, to construct Product P is called a **component** of the bill-of-material relationship. The parent represents the **end point** and the component represents the **starting point** of the bill-of-material relationship. This type of bill-of-material relationship can be identified by the combination of the product code of the parent and the product code of the component. One of the properties of a bill-of-material relationship is that it is transitive. This means that if B is a component of Product A and C is a component of B, then C is an (indirect) component of Product A. A **parts list** is defined as a list of all of the bill-of-material relationships that have a common parent. This list consists of all of the product codes of the direct components of a given product.

The **quantity-per** is a required attribute of each bill-of-material relationship. This quantity-per specifies the number of units of the component needed to build one unit of the parent product.

In some instances it is necessary to specify the sequence in which a product must be constructed from its components. This can be documented by adding a **sequence number** attribute to the bill-of-material relationship. In this case the bill-of-material relationship that was identified by the combination (parent product code, component product code) must now be identified by (parent product code, sequence number, component product code). This approach is useful particularly when the sequence number can also be used as the basis for sequencing the production operations. This means that the sequence number can be interpreted as a major task comprised of a number of lower level operations. The sequence number is often referred to as a "pos number" (short for "position number").

This situation can be illustrated using the example of a desk chair consisting of a underframe, a seat, a back and two armrests. The product structure of this desk chair is presented in Figure 4.2. The underframe and seat components are required at sequence number 1. Two armrests are required as components to complete the construction of the desk chair at sequence number 3.

The product and bill-of-material entities have been completed for this desk chair example in Figure 4.3.

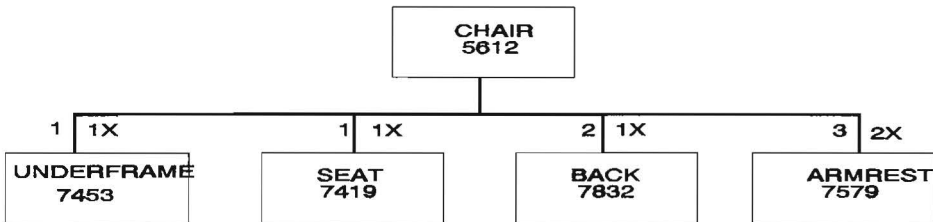


Figure 4.2: Product structure of a desk chair

The finished product in this example is the desk chair. All of the other products are primary products.

The desk chair in this example is constructed in a single stage (consisting of three major tasks). This example can be extended by specifying that the underframe is assembled from a stand and five wheels. Two product descriptions for the stand and the wheels must then be added to the product entity type to document this change. This results in the addition of two new bill-of-material relationships representing the underframe/stand relationship and the underframe/wheel relationship.

A product may appear as a component in a variety of different bill-of-material relationships with different parents. This may lead to a network of products and bill-of-material relationships within a company. This network can be represented as a acyclic, directed graph. A **bill-of-material** for Product P, thus, becomes a sub-graph in which all of the products included in Product P appear as nodes.

Products can be identified in two ways. One way, referred to here as the **direct method of identification**, is identification via the product code. The second way, the indirect method of identification, is typically used for a range of products for which there are a large number of more or less similar product variants. In this case an enormous number of product codes would be needed to refer to all of the possible product variants. This implies that the task of maintaining all of these product codes would also be enormous. The **indirect method of identification** means that a finished product is described based upon a list of components. Instead

of using a single product code to identify products via the direct method, a list of product codes and the way in which a product is constructed is used to identify a product via the indirect method. An example of this would be using the component series of {7419, 7453, 7579, 7832} to specify a certain desk chair instead of the product code of 5612 assigned to this chair. It is assumed that the way in which the components are used to construct this chair is known, e.g., it always consists of a underframe, a seat, a back and two armrests. Each of these components must be specified when using the indirect method of identification.

entity type: PRODUCT		
product code	unit	description
5612	piece	desk chair
7419	piece	seat
7453	piece	underframe
7579	piece	armrest
7832	piece	back

entity type: Bill-of-material RELATIONSHIP			
parent product code	sequencen umber	component product code	bill-of- material quantity
5612	1	7419	1
5612	1	7453	1
5612	2	7832	1
5612	3	7579	2

Figure 4.3: Bill-of-material for the desk chair

### Presentation

A bill-of-material can be presented in several different ways. There are two basic forms: an exploded list and an imploded list. An **exploded list** of a product's bill-of-material provides a list of the components at one or more levels of the product structure. An **imploded list**, on the other hand, provides a specification of the parents of a product. Three different types of exploded lists can be identified:

1. a single level explosion in which all of the direct components of a given product are specified;
2. a multiple level explosion in which the direct and indirect components of a given product are specified, sorted by level;
3. a total explosion in which all of the primary products used as components for a given product are specified.

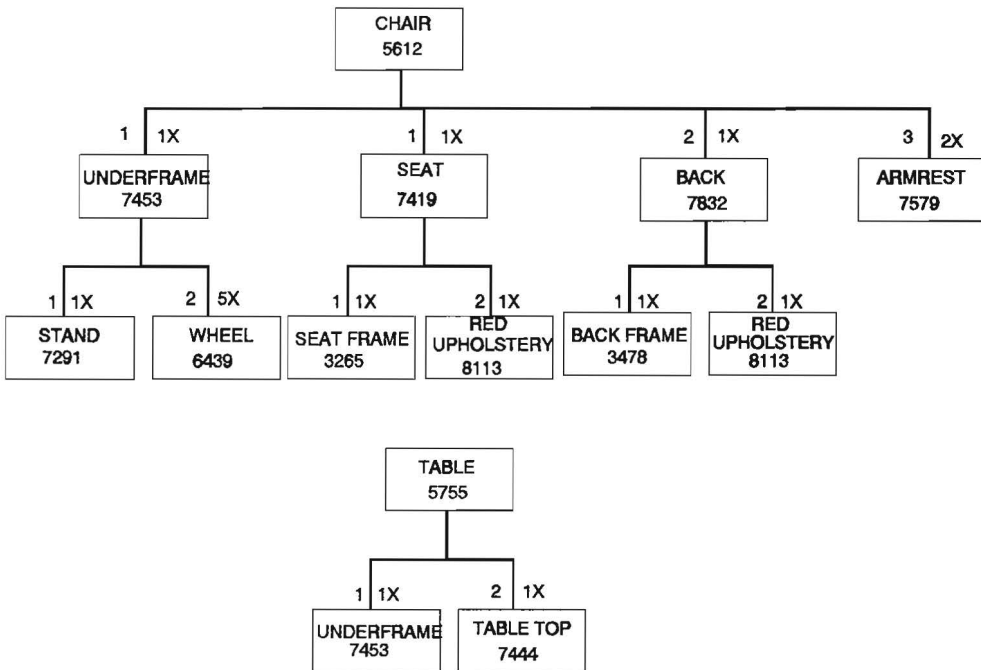


Figure 4.4: Product structures of the desk chair and the table

entity type:      PRODUCT		
product code	unit	description
3265	piece	seat frame
3478	piece	back frame
5612	piece	desk chair
5755	piece	table
6439	piece	wheel
7291	piece	stand
7419	piece	seat
7444	piece	table top
7453	piece	underframe
7579	piece	armrest
7832	piece	back
8113	m2	upholstery

entity type:      Bill-of-material RELATIONSHIP			
parent product code	sequence number	component product code	quantity-per
5612	1	7419	1
5612	1	7453	1
5612	2	7832	1
5612	3	7579	2
5755	1	7453	1
5755	2	7444	1
7419	1	3265	1
7419	2	8113	0.5
7453	1	7291	1
7453	2	6439	5
7832	1	3478	1
7832	2	8113	0.3

Figure 4.5: Bill-of-material list

A single level explosion represents a parts list as defined above, while a multiple level explosion generates a bill-of-material that also lists indirect components.

A number of examples of bill-of-material explosions are presented here based upon an expanded version of the bill-of-material presented in Figure 4.3 with a finished product called "table" and a product consisting of a underframe, seat and back (see Figure 4.4). The bill-of-material list presented in Figure 4.5 is used as the basis for the different forms of explosions shown in Figure 4.6 for the chair product defined in Figure 4.4.

Product code: 5612 chair			
sequence number	component number	description	quantity
1	7419	seat	1
1	7453	underframe	1
2	7832	back	1
3	7579	armrest	2

single level explosion

Product code: 5612 chair		
primary product code	description	quantity
3265	seat frame	1
3478	back frame	1
6439	wheel	5
7291	stand	1
7579	armrest	2
8113	upholstery	0.8

total explosion



Product number: 5612		chair	
sequence number	component number	description	quantity
1	7419	seat	1
. 1	3265	seat frame	1
. 2	8113	upholstery	0.5
1	7453	underframe	1
. 1	7291	stand	1
. 2	6439	wheel	5
2	7832	back	1
. 1	3478	back frame	1
. 2	8113	upholstery	0.3
3	7579	armrest	2

multiple level explosion

Figure 4.6: Different explosions for a product

We can produce similar tables for the three types of implosions: a single level, a multiple level and a total implosion. The direct parents, all of the direct and indirect parents and all of the highest level products are shown, respectively, for a given product in this case.

### Path

A **path** in a bill-of-material list is an ordered list of bill-of-material relationships in which the first relationship of each sequential pair of bill-of-material relationships represents a component with its parent as the second relationship. Since a bill-of-material relationship has a direction from component to parent in this way, the path can be viewed as a directed path. If a path leads from Product A to Product B, then A can be seen as a component (possibly with multiple levels) of B and B can be seen as a parent (possibly with multiple levels) of A.

An example of this can be found in Figure 4.4 where two paths exist between the chair product and the upholstery product in the bill-of-material list. The first path

is defined by two bill-of-material relationships which can be coded as 7419-2-8113 (parent - sequence number - component) and 5612-1-7419. The second path is defined by the bill-of-material relationships 7832-2-8113 and 5612-2-7832. These two paths also can be identified easily in the multiple level explosion presented in Figure 4.5.

The **length** of a path is defined as the number of bill-of-material relationships found in that path. The **level** of a given Product A appearing within a bill-of-material for Product B is defined as being the maximum path length between A and B. Multiple paths may lead from A to B. This is apparent in the example where multiple paths exist between the chair and the upholstery. The **lowest level code** of a given Product A is defined as the maximum value found among all of the levels of Product A appearing on the bill-of-material list. The lowest level code can also be seen as the length of the longest path between a highest level product and Product A. A highest level product, thus, always has a lowest level code of zero. The upholstery product in our example has a lowest level code of two. The lowest level code of a component in a bill-of-material relationship is always at least one greater than the lowest level code of its parent. The lowest level code is useful when calculating the cost price of a product and determining material requirements.

### **Engineering changes**

A engineering change to a product may be required for a number of reasons such as a product improvement, market demand, a customer requirement or a change in the manufacturing process [KNOX84]. Two essential concepts need to be defined more precisely before we can describe the different types of engineering changes, however.

The concept of **interchangeability** is the first aspect that needs to be defined. Every organization has its own idea of what is meant by interchangeability. The definition of product interchangeability used here is that products are interchangeable when they are equivalent with respect to form, fit and function in all of the subassemblies and finished products in which these products are used.

---

The concept of a **version** of a product is the second aspect that needs further clarification. Whenever a product is modified and, in addition, the modified and original products are interchangeable and can be substituted for each other, then the modified product is referred to as a version of the original product. The **version number** is used to keep track of the number of versions that exist for any given product. This version number is an attribute of the product entity. The version number may not be included as part of the product identification key since the different versions of a product are interchangeable. The different versions of any given product are treated as being equivalent as far as the end user is concerned.

The following types of technical modifications can be identified:

1. product modifications where the original and modified products are interchangeable. This type of modification does not have any impact on the end user's situation; the creation of a new (higher) version number is the only noticeable difference.
2. product modifications where the original and modified products are not always interchangeable without taking certain factors into account, but where the respective parent products are interchangeable. The consequence of this is that a new product code must be assigned to the modified product and that a new version of one or more parent products needs to be created.
3. product modifications where the original and modified products are not always interchangeable without taking certain factors into account, and where the respective parent products are also not interchangeable. The consequence of this, to start with, is that new product codes must be assigned to the modified product and its parent product(s). Subsequently, with respect to the higher level products that are parents of the modified product in the bill-of-material set, an analysis must be made to determine the degree to which these products can be substituted for each other. This means that the path must be followed from the modified product to a higher level product until a level is reached where product substitution is possible. The top of the modification path is represented by the bill-of-material relationship in this modification path where the parent product can be substituted but the component product cannot be substituted.

When a product modification leads to the assignment of a new product code, this will be referred to as a product code modification in the remainder of this thesis. A product code modification is always associated with a specific date upon which the change is to become effective. The new component will be substituted for the original component in a subassembly as of this date. This type of change can be noted in the parts list by indicating that the bill-of-material relationship of the original component has expired and that the bill-of-material relationship of the new component is currently in effect. In order to record this type of modification, each of the bill-of-material relationships must have the following attributes: an effective date, a change order number associated with the effective date, an expiration date and a change order number associated with the expiration date. It is important to note that all of the bill-of-material relationships associated with a given change order number may have different dates.

When the aforementioned attributes are added to the bill-of-material relationships, a validity date needs to be specified whenever an application uses the bill-of-material relationship data so that the product codes can be selected which will be in effect on that date. If, for example, a total explosion is to be generated for a specific product, then the product code as well as a date need to be specified.

It is relevant to consider which bill-of-material relationships are affected when a product change is made. There are two possibilities. The specification of an expiration date and the associated change order number is required at the highest level relationships in the modification paths. The specification of an effective date and the associated change order number is required for all of the newly substituted bill-of-material relationships.

### **Data model**

The following data model is needed to record the aforementioned data in a proper manner. This data model specifies which entity types are generally required to represent the product information within a given company:

- . product and product identifier (product code);
- . bill-of-material relationship and identifier  
(parent code, sequence number, component code);

- . parent sequence and identifier (parent code, sequence number);
- . change order and identifier (change order number).

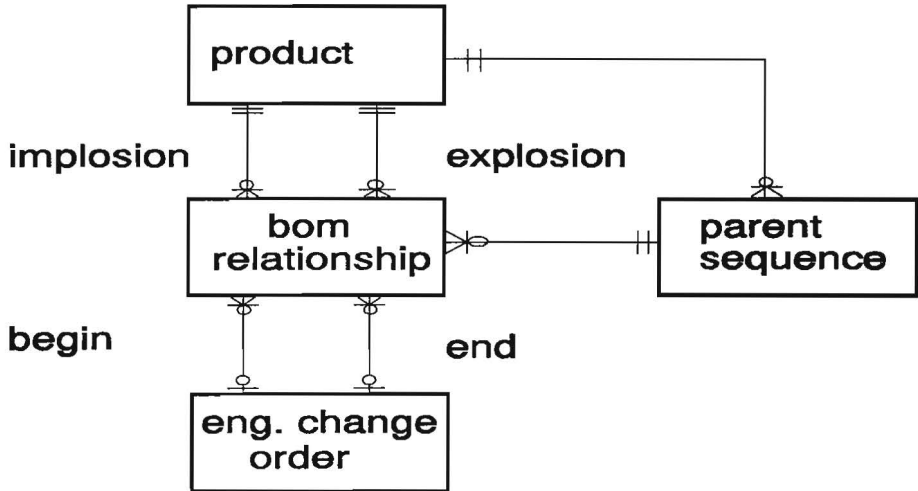


Figure 4.7: Data structure diagram for a bill-of-materials system

Figure 4.6 shows the interrelationships between these different entity types in the form of a data structure diagram. The basic attributes needed to describe each of these entity types are as follows:

- . the product code, unit of measurement, description and version number attributes for the product entity type;
- . the parent code, sequence number, component code, quantity-per, effective date, change order number associated with the effective date, expiration date and change order number associated with the expiration date attributes for the bill-of-material relationship entity type;
- . the parent code and sequence number attributes for the parent sequence entity type;
- . the change order number and description attributes for the change order entity type.

**Points of view**

A bill-of-material for a finished product can be produced in a variety of different forms, depending upon the way in which the person generating the bill-of-material views the product. A product designer, for example, will typically view a product in terms of components that correspond to specific functions which play a role in designing a product [VEEN90]. Van Rijn has performed an in-depth study of the requirements of the various business functions related to the specification of a complete and correct product description in the form of a bill-of-material [RIJN85].

Various types of bills-of-materials may exist for a given finished product that differ with respect to how subassemblies are defined and used. This typically leads to different definitions for the product entity type as well as for the bill-of-material relationship entity type. For example, a bill-of-material used for logistic and distribution functions will define subassemblies based upon the products held in stock or needed for planning purposes.

## 5 Basis for the generic bill-of-material

---

### Introduction

A definition for a bill-of-material was provided in Chapter 4 and several aspects related to bills-of-materials were discussed. This was summarized in terms of a data model for a bill-of-material. This chapter covers the subject of documenting product knowledge for a set of similar products using the concept of a generic bill-of-material. We start here by classifying the existing documentation systems and then continue by describing the concept of a generic bill-of-material in detail. This concept is subsequently used in the remainder of this thesis. This chapter concludes with the description of a data model for this generic bill-of-material system.

### Product family

A group of products can be defined as a **product family**. A product belonging to such a family is referred to as a **variant** within this product family. A product family is usually defined based upon a standard product. This means that the differences in the product structures of the various variants within a product family will be minimal.

### Identifying a variant

**Identification** is defined here as the determination of a product's identity. There are several known methods that can be used to define and identify a variant within a product family. These methods are classified here in two different ways.

One method of classification is to make a distinction between a **direct** versus an **indirect** method of identification. A variant can be defined and identified directly by assigning a unique product code to it. An indirect method of identification is to describe a variant through the use of a bill-of-material or a partial bill-of-material (refer also to Chapter 4).

This can be illustrated by expanding the example of the chair (see Figure 4.5). Two variants of this chair can be provided: a red chair or a blue chair. The difference

between these two variants is found in the different colors of upholstery used for covering the seat and the back.

Just the use of a product code is sufficient for directly defining and identifying a particular variant of this chair. For example, the existing product code of 5612 could be used to identify a red chair and a new product code of 5613 could be established to identify a blue chair. The red chair could be identified indirectly through the use of the parts list. An example of such a parts list is presented in Figure 5.1.

This type of approach is used in standard MRP-II software packages to identify the variants of finished products [VEEN91]. When the product family is already known, however, it is sufficient for the purpose of product identification to specify only the components in the parts list which vary within the product family. To illustrate this, assume that the chair in Figure 5.1 is the red variant within a product family of chairs for which the variants are determined based upon the color of the seat and back. This means that it is sufficient to specify only that the seat is red and the back is red to uniquely identify this product variant within this family. A similar identification could be used to record a customer order. This form of identification could be used in combination with the bill-of-material system to generate a complete bill-of-material for a specific customer order. This is then sufficient information for filling the customer order.

Wedekind and Müller [WEDE81] make use of the indirect method of identification for describing a variant within a product family.



Product number: 5612 chair			
sequence number	component number	description	quantity
1	7419	red seat	1
1	7453	underframe	1
2	7832	red back	1
3	7579	armrest	2

Figure 5.1: Presentation of an indirect identification of a red chair

A second method of classifying descriptions of product variants is through the use of **parameters**. This implies that a list of parameters is associated with each product family. Each of these parameters subsequently has a list of valid parameter values.

The chair product family has only one parameter in our example: color. The list of valid parameter values associated with this parameter contains two entries, red and blue. The identification of a red chair could then be formulated as 5612G[red]. The product code 5612G in this example represents the chair product family.

Schönsleben [SCHö85], Van Veen [VEEN91], Carruthers [CARR76] and the SCOUT [DANI] software package all make use of parameter descriptions in this way to identify product variants.

### Bill-of-material systems for product families

A **product family bill-of-material (PFBOM)** is a model of a product family through which a bill-of-material for a product variant can be found by specifying a variant identification code. A **product family bill-of-material set** is the collection of all of the product family bills-of-materials within a given company. A third term that is important in this context is the **product family bill-of-material**

**system.** This term is defined here as an automated information system used to create and maintain a product family bill-of-material set.

In addition to these two methods of classification used to identify product variants (direct versus indirect, parameter driven), a third method also exists. It is important to mention this third approach in order to provide a complete picture of the various types of product family bill-of-material systems that are available. Before this third approach is described, however, it is important to note that the aforementioned PFBOM systems are capable of defining product families only at the finished product level. This is mentioned because the third method of classification is an approach in which PFBOM systems are divided into two categories: those with (recursive) and those without (non-recursive) the capability of defining product families at the component level. The first category is, thus, comprised of systems that employ the same method of identification for product families at the component level as for product families at the finished product level. A finished product variant must also be able to include components that are, in turn, defined as variants within a product family of components. Van Veen, for example, has developed a system that fits this description [VEEN91].

A classification of the aforementioned systems according to the various characteristics as previously described is presented in Figure 5.2. (refer to Chapter 10 for a further description of the systems)

The PFBOM, or generic bill-of-material, as described below can be classified in three different categories in the most complex situation:

1. a PFBOM with indirect identification;
2. a PFBOM with parameter identification;
3. a PFBOM with product families defined at the finished product level as well as at the component level.

Identification		product family	
		non-recursive	recursive
direct	without parameters		product code
	with parameters	Carruthers SCOUT	Van Veen
indirect	without parameters	MRP-II	Wedekind & Muller
	with parameters	Options & features	

Figure 5.2: Classification of product family bill-of-material systems

**Generic product**

A **generic product (GP)** is a set of products. A generic product may represent a set of finished products, a set of primary products or a set of subassemblies. A generic product may cover one or more actual products. The construction of the products associated with a generic product should be more or less similar in order to be used effectively as part of a requirement specification.

Since the construction of a product in our point of view is specified primarily via the bill-of-material, all of the products associated with a given generic product need to have bills-of-materials that are more or less similar. These similar bills-of-materials are generalized here by introducing a new concept, the **generic bill-of-material**. A generic bill-of-material is a bill-of-material for a whole product family. Such a generic bill-of-material can be represented by two entity types, namely: a generic product and a generic bill-of-material relationship.

The generic bill-of-material relationship entity type provides the basis for documentation the relationship between a generic parent and a generic component. This type of relationship implies that **at least one** bill-of-material relationship must exist between one of the parents associated with the generic parent of the generic bill-of-material relationship and one of the components associated with the generic component of the same generic bill-of-material relationship. This is not consistent with the approach used by Van Veen for defining a generic bill-of-material relationship [VEEN91, page 108]. Van Veen assumes that there must always be a relationship for every parent associated with the generic parent and at least one of the components associated with the generic component of the same generic bill-of-material relationship.

### **Basic concept**

Each variant of a generic product is constructed in more or less the same way. The differences between the generic product variants become apparent only after the generic components of the variants are specified. The generic components differ, in turn, with respect to their component variants. Ultimately, we can trace the differences between the product variants to the variants of the generic products at the lowest level in the generic bill-of-material: the generic primary products.

The basic concept of a generic bill-of-material can, therefore, be formulated in the following way. The variation within a generic Product G can be traced to the variation in the generic primary products at the beginning of the paths that lead from the generic primary products to the generic Product G in the generic bill-of-material. The implications of this basic concept will be explained further in the remainder of this chapter.

### **Generic primary product**

The generic primary products are found at the lowest level of a generic bill-of-material set. A generic primary product normally incorporates a number of variants of a primary product, such as the upholstery in our example of the chair. A **generic primary product** (GPP), therefore, is defined here as the set of all of the variants

---

of a primary product. The GPP called "upholstery" can be represented by the set {red upholstery, blue upholstery}.

The variants within a GPP can be identified in a number of ways, specifically: directly or indirectly and, optionally, through the use of parameters. The direct identification of the variants within a GPP can be accomplished through the use of product codes. Another method of describing a variant within a GPP is through the use of parameters. This means that a list of parameters is associated with each GPP and a list of the valid parameter values is associated with each parameter. A GPP variant can be identified for each valid combination of parameter values.

For example, the GPP called "upholstery" can be described using the parameter "COLOR" which has one of the values from the set {RED, BLUE}.

It is conceivable that a GPP will be defined with only one possible variant. This means that the use of parameters would not be necessary and the product code of the primary product could be the same as the product code of the GPP. In the example of the seat frame that has no variants, it can be seen that the GPP called "seat frame" has only one element.

The question of how to record the valid combinations of parameter values will be discussed later.

The only restriction placed on the parameter descriptions for a GPP is that after all of the parameter values have been assigned, each combination specifies one and only one product variant. This restriction does not apply to the non-primary generic products, however.

### **Generic subassembly**

A **generic subassembly** (GS) consists of subassemblies that are constructed based upon more or less the same bill-of-material. A product code can be assigned to each variant within a generic subassembly for the purpose of identifying these variants.

The variation of products found within a GS is directly related to the variation of products in the associated GPP's since the bills-of-materials for all of the GS variants are more or less the same. This suggests that a different approach to identification, indirect identification, might be better. This can be illustrated using the example of a red seat for a chair.

The basic question for identifying a variant within a GS in this example is determining the way in which the red seat is different from the blue seat. The answer to this question is that one of the primary products represents the difference between these two subassembly variants, namely, the upholstery. The red seat uses red upholstery and the blue seat uses blue upholstery. This means that in order to specify the red seat, it is sufficient to indicate that this is a variant of the "seat" GS that includes the red variant of the "upholstery" GPP. In other words, the indirect identification and the parameter identification can be combined. This combination is used to identify a GS variant in this case. The way in which this type of identification can be represented is illustrated in Figure 5.3a for the red seat. The generic products are coded using four numbers and the letter "G (for Generic). It is necessary to specify the path that leads from the GPP to the GS for an indirect identification. This is illustrated in Figure 5.3a under the heading "sequence number".

Within the context of the generic product family called "chair," it is not necessary to specify a seat frame in order to identify a seat variant since the seat frame is required to make all of the seats (Figure 5.3b). This type of generic component is referred to as a common component. Such common components will not be included as part of the identification in the subsequent discussions and examples.

sequence number	generic product number	description	parameters	parameter value
1	7419G	seat		
. 1	3265G	seat frame		
. 2	8113G	upholstery	COLOR	RED

Figure 5.3a: Identification of a red seat

sequence number	generic product number	description	parameters	parameter value
1	7419G	seat		
. 2	8113G	upholstery	COLOR	RED

Figure 5.3b: Identification of a red seat  
(without common generic components)

### Paths

Using the following example, we can see that it is necessary to make use of paths for identifying a product variant (Figure 5.4). This example is the identification of a chair with a red seat and a blue back. The generic bill-of-material for this chair can be exploded into multiple levels as shown in Figure 5.4. As shown here, the product codes are translated into generic codes by appending the letter "G." Since the "upholstery" GPP appears twice in this bill-of-material, two separate specifications are needed to indicate which upholstery is to be used for the seat and which upholstery is to be used for the back. In this example the red upholstery is chosen for the seat and the blue upholstery is chosen for the back. This means that two paths need to be included in Figure 5.4 to specify a chair with a red seat and a blue back.

sequence number	generic product number	description	parameters	parameter value
1	5612G	chair		
. 1	7419G	seat		
. . 2	8113G	upholstery	COLOR	RED
. 2	7832G	back		
. . 2	8113G	upholstery	COLOR	BLUE

Figure 5.4: Identification of a chair with a red seat and blue back

The specification of a GS variant is considered to be complete when all of the GPP's at the beginning of each of the paths have been specified.

### **GPP coordination**

It is often the case that the GPP's belonging to a GS have the same parameters. In some instances all of the parameter values for these parameters must be the same for a given GS variant. This can be achieved by coordinating the values of these parameters via a common generic parent of the relevant GPP's. This type of coordination is referred to as **GPP coordination**.

Returning to our example, it is possible to specify that the color of the seat and the color of the back must always be the same. This means that it should only be necessary to specify the desired color only once instead of twice. This can be achieved by using a common parent of the seat upholstery and the back upholstery to record this specification. The common parent in this case is the "chair" GS that subsequently needs to have a parameter called COLOR. The COLOR parameter, thus, becomes a parameter for the chair as well as a parameter for the upholstery.

### **Inheritance**

When a parameter is defined and used by a common parent to coordinate several GPP's, the chosen parameter value must be passed down to the relevant GPP's. The transfer of parameter values in this way is called **inheritance**. All of the GS's that are included in a path between the GPP's to be coordinated and the common parent must have this coordinating parameter. All of these GS's will then inherit the relevant parameter value.

If we apply this concept to the example above, the "seat" and "back" GS's will both have the COLOR parameter since they are included in either the upholstery-seat-chair path or the upholstery-back-chair path. It is then quite simple to identify and specify a chair with red upholstery (see Figure 5.5).



sequence number	generic product number	description	parameters	parameter value
1	5612G	chair	COLOR	RED

Figure 5.5: Identification of a red chair

It is advisable to define a common parent at a level that is as low as possible in the bill-of-material structure in order to minimize the number of times that extra data such as parameters will need to be recorded for GS's that are involved in inheriting parameter values.

The coordination of GPP's does not necessarily need to deal with only parameters that are identical. It is possible that parameters at a higher level in the bill-of-material will need to be translated into different parameters at the GPP level. This type of translation process will be discussed in the following chapter.

Introducing GPP coordination typically results in:

- . reducing the total number of possible variants;
- . identifying products at a higher level in the generic bill-of-material;
- . reducing the number of parameter values that need to be specified.

### Specialties associated with a generic product

A generic product is a set of products. The elements of such a set are, thus, the product variants. When one or more parameters are assigned to a generic product, it is convenient to identify various subsets of product variants. Such subsets are referred to as products with **specialties** that are defined by specifying the relevant parameter values for these subsets. A specialty may refer to an empty set, a "singlet" (single variant) or a "tuple" (multiple variants). In a situation in which every finished product can be identified and specified using a generic bill-of-material, then a specialty for a generic primary product will always refer to either an empty set or a singlet.

If values have been assigned to all of the parameters for a generic subassembly, then this does not necessarily mean that a single variant has been specified. A subset that has been specified in this way will normally have multiple variants due to the likelihood that additional parameter values will still need to be specified at lower levels in the generic bill-of-material.

To illustrate this we can expand on the example of the chair once more. If we assume that the chair can be produced in a variant with wheels and a variant without wheels as well as a variant which swivels and a variant which does not swivel. This can be accomplished by defining a number of variants for the stand and for the wheels.

Each wheel can be viewed as a generic primary product met de parameter WHEELS and the possible parameter values of YES and NO. The use of this parameter creates two subsets, namely: a singlet (i.e., with wheels) and an empty set (i.e., without wheels). Two parameters are assigned to the stand: WHEELS and SWIVEL. The permissible values for both of these parameters are YES and NO.

The WHEELS parameter should be coordinated by the lowest common parent of the wheel and the stand: the underframe. As a result, the WHEELS parameter is also assigned to the underframe so that we can now identify two specialties: underframes with wheels and underframes without wheels. Each of these specialties has two variants: with swivel and without swivel. The SWIVEL parameter is not defined at the underframe level since this parameter is only applicable to the stand. The value of the WHEELS parameter will be inherited from the underframe to the wheel and to the stand.

The specialties belonging to the GPP's that are used to identify a variant are either empty sets or singlets. If this is not the case, then the identification is considered to be **incomplete**. An incomplete identification may be useful during the design phase of a product's life-cycle.

sequence number	generic product number	description	parameters	parameter value
1	5612G	chair	COLOR	BLUE
. 1	7453G	underframe	WHEELS	YES
. . 2	7291G	stand	WHEELS	YES
			SWIVEL	YES

Figure 5.6: Identification of a blue, swivel chair with wheels

The example described above is illustrated in Figure 5.6 for a blue swivel chair with wheels. Note that a specialty is specified on each line of the identification. When an identification path is followed from a GS to a GPP, then the number of GPP variants for a specialty on a given path will either decrease or remain the same. We see the following when we look at the number of variants for the stand on the path shown in Figure 5.6. The first specialty on this path is the blue chair. This standard blue chair option includes all four of the stand variants. The second specialty, the underframe with wheels, reduces the number of stand variants to two: wheels with swivel and wheels without swivel. The third specialty ultimately provides only a single variant: wheels with swivel.

### Forms of identification

Three different forms of identification have been used within the generic bill-of-material system. Figure 5.7 shows how the identification of a number of the chair parts can be classified. These various forms of identification can also be used to describe a chair.

When a specific variant of a chair is identified directly without parameters, then a unique product code is used. This means that chairs are not described in terms of variants of a product family. This also affects the generic components of the chair. Since the generic chair product is a singlet, all of its components must also be defined as singlets. No other choices can be made at the lower levels in the bill-of-material since this would mean that multiple variants are possible.

When a specific variant of a chair is identified directly without parameters, then a unique product code is used. This means that chairs are not described in terms of variants of a product family. This also affects the generic components of the chair. Since the generic chair product is a singlet, all of its components must also be defined as singlets. No other choices can be made at the lower levels in the bill-of-material since this would mean that multiple variants are possible.

Identification within a product family		
direct	without parameters	seat frame back frame
	with parameters	upholstery seat
indirect	without parameters	
	with parameters	chair underframe

Figure 5.7: Forms of identification within the generic bill-of-material

A specific chair that is identified directly with parameters is described in terms of parameters at the highest level in the bill-of-material. All of the parameters that can be specified in the original bill-of-material are determined at the chair level.

There is an inherent problem with this approach. This problem occurs when a certain generic Component C appears in multiple paths in the generic bill-of-material and different variants of this Component C need to be specified. This means that the component parameters then need to be defined more than once for the high-level product. In addition, a different set of parameter values must be maintained for each path in which this component is used. This can be visualized by referring to the example of the chair with a red seat and a blue back in Figure 5.4. The generic upholstery product appears twice in the generic bill-of-material for the chair. The color parameter for the upholstery will need to be defined twice for the generic chair product in this situation: once to specify the color of the seat upholstery and again for the color of the back upholstery.

The identification of the components must be made directly, with or without parameters, when the highest level product is specified directly with parameters.

Figure 5.8 shows the six combinations of parents and components that are possible within the context of the three forms of identification of parents and the three forms of identification of components.

			P A R E N T		
			direct		indirect
			without	with	with
C O M P O N E N T	direct	parameters			
		without	valid	valid	valid
	with		valid	valid	
indirect	with			valid	

Figure 5.8: Valid parent/component combinations

### Parameters

A classification of the parameters needed to identify a generic product is presented in this section.

Each generic component may inherit certain parameter values from its parent. The associated parameters are referred to as the **external parameters** of the generic component. In the example of the chair used here, the WHEELS parameter is external for the stand. A specialty that is characterized by a valid combination of inherited parameter values is called a **top level specialty**. This means that all of the parameters associated with a top level specialty are external parameters. For

component of numerous variants of the same generic parent. This means that this variant of a generic component may be included in several top level specialties. This is because every variant of a generic parent can transfer its parameter values to only one top level specialty. As a result, different variants of a generic parent can be associated with different top level specialties for the generic component.

This is illustrated in Figure 5.9 using a slightly modified version of our chair example. As we have seen previously, each chair variant is comprised of a single underframe variant and other components. The generic chair parent is a set with three elements: the red, the gray and the blue variants. The generic underframe component is a set with four elements: the swivelling variant with wheels, the non-swivelling variant with wheels, the swivelling variant without wheels and the non-swivelling variant without wheels.

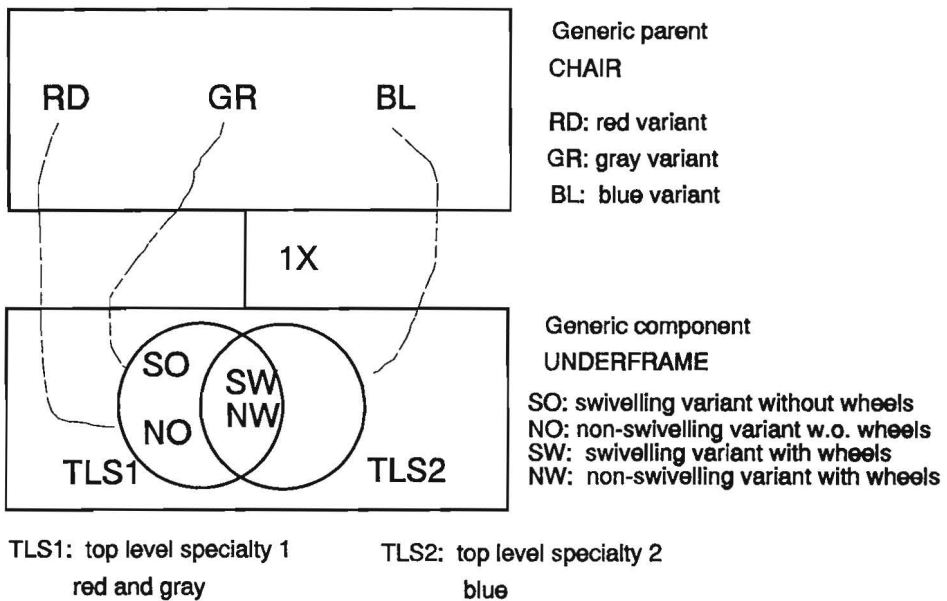


Figure 5.9: Mutually dependent high-level options

For the red and gray variants of the chair, all of the underframe variants can be chosen. These underframe variants represent the subset of options for the generic

---

underframe component. This subset can be characterized by two external parameter values: red and gray. This subset can, thus, be seen as a top level specialty (TLS1).

For the blue chair variant, however, the choice is limited to the two variants with wheels. These two variants are represented by a top level specialty (TLS2) of the generic underframe component with the color blue. This example shows that the two underframe variants with wheels belong to top level specialty TLS1 as well as top level specialty TLS2. The intersection of these two subsets is not empty. In other words, these two top level specialties are not mutually independent.

The external parameters inherit their parameter values as part of the identification process. The parameter values for the remaining parameters of a GP must be chosen at the GP level. These parameters are referred to as **own parameters** of the GP. The own parameter for the stand, for example, is the SWIVEL parameter. The external and own parameters, together, represent the complete set of parameters for a GP.

There are a number of different ways to choose the values for the own parameters. An initial assumption is that this decision process will be structured as a decision tree. (Other alternatives will be discussed later.) The nodes of the decision tree are the specialties. As we have seen, the specialties can be viewed as sets of product variants.

The decision process starts with a top level specialty. This can be visualized at the root of the decision tree. Next, a choice is made from a number of subsets that, together, represent all of the variants of the top level specialty. In addition, these subsets are mutually independent within a given top level specialty. The top level specialty is essentially split into a number of subsets in this way. In other words, each variant of the top level specialty is found in one of the subsets of this option. Each of these subsets can be seen as a specialty that can be characterized in terms of parameter values. Minimally, there will be at least one own parameter. The branches of the decision tree represent the relationships between the generic product and the various specialties. This means that the decision tree can be viewed as a hierarchical representation of the generic product structure.

The leaves of the decision tree represent the **lowest level specialties** at the lowest level for the generic product. All of the lowest level specialties, taken together, include all of the variants of the generic product and are mutually independent. A lowest level specialty for a generic primary product consists of either a single variant or no variant. The parameters of a lowest level specialty are referred to as the **internal parameters** of the generic product.

An internal parameter may be an external parameter as well as a own parameter. If the internal parameters are the same as the external parameters for a given GP, then this GP can be identified directly through the use of these parameters. In this case the GP will, in fact, have no own parameters. In other words, no decision processes are required to determine a variant of this GP at the GP level. Similarly, there is no decision tree. The top level specialties are the same as the lowest level specialties. The seat in our illustration of the chair is an example of this. The COLOR parameter for the seat is an external parameter as well as an internal parameter. The opposite case in which the internal parameters are the same as the own parameters will be described later in this section.

The stand GPP in our example of the chair has four lowest level specialties with internal parameters called WHEELS and SWIVEL. A generic product variant may occur in connection with a top level specialty as well as a lowest level specialty. An example of this is the variant of the stand that swivels and has wheels. This variant is associated with the top level specialty of a stand with wheels as well as with the lowest level specialty of a stand that swivels and has wheels.

A lowest level specialty may also be the same as a top level specialty. The external parameters and internal parameters are identical in this case. This type of specialty essentially transfers the parameter values that it has inherited from its parent to any components it may have. The seat and the back in our example both have this type of specialty in the form of the color parameter. The parameter value for the color is inherited from their parent, the chair. This parameter value is subsequently passed on to one of their components, the upholstery.

All of the variants of a GP may appear as components in each of the parent variants when a GP with parameters does not inherit any parameter values. Since the GP in this case has parameters, one or more parameter values will need to be



selected. In other words, a decision process must take place. Every decision process associated with the generic bill-of-material is based upon a decision tree. A top level specialty is found at the top of the decision tree. This top level specialty includes all of the valid variants of the GP. This type of specialty is called a **universal top level specialty** and has no parameters associated with it. The internal parameters are the same as the own parameters in the case of a GP with a universal top level specialty.

The underframe in our example of the chair can be used to illustrate the concept of a decision tree. As seen previously, two specialties are associated with the underframe. One option is represented by the underframe variants with wheels and the other option is represented by the underframe variants without wheels. The value of the WHEELS parameter is not inherited. This means that this parameter is a own parameter. No parameter values are inherited by the chair underframe. Thus, by definition, the underframe only has one top level specialty, namely, a universal top level specialty. This is illustrated in the upper part of Figure 5.10.

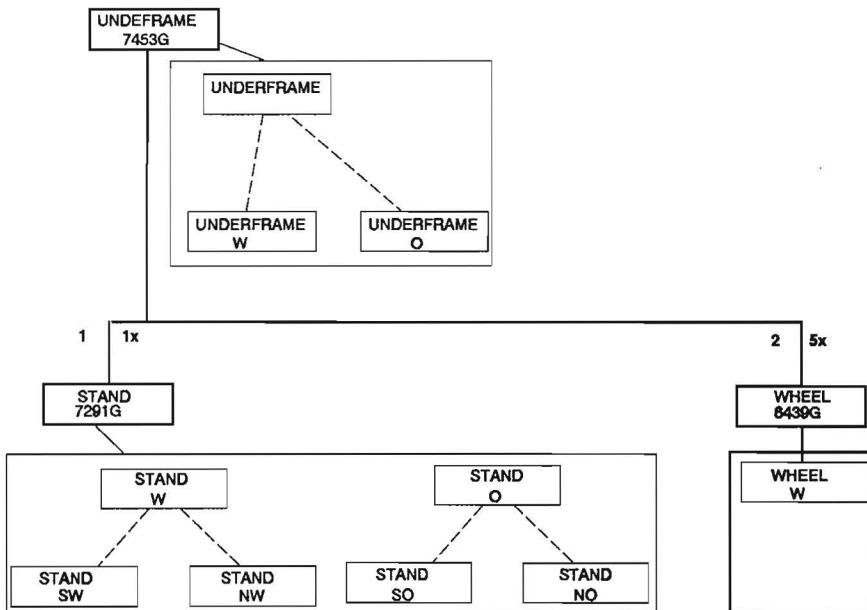


Figure 5.10: Generic bill-of-materials for the frame

The decision tree for the underframe components, the stand and the wheels, are also shown in Figure 5.10. Two top level specialties are associated with the stand. These top level specialties are characterized by the inherited WHEELS parameter. Each top level specialty can be represented as the root of a decision tree. For both of the top level specialties, a choice must be made between a swivelling and non-swivelling variant. In this way there are ultimately four lowest level specialties with respect to the stand.

The generic wheel product contains a single top level specialty that is described by the WHEELS parameter with a value of YES. The other possible top level specialty represents the variants without wheels, but this is an empty specialty. This empty specialty can be suppressed in the generic bill-of-material since it does not need to appear in the decision tree. A complete generic bill-of-material, therefore, consists of a bill-of-material structure and a decision tree structure.

### **Prerequisite conditions for inheritance**

After introducing the concept of a decision tree, we are now able to discuss the subject of inheritance in more detail. The inheritance process transfers information from the parent to its component. This information may relate to quantities for materials planning or could otherwise deal with parameter values. We will limit the scope of our discussion here to the second type of information.

The parameter values that can be inherited from a parent are associated with the internal parameters of that parent. In other words, these are the parameters associated with the lowest level specialties of the parent. Not all of the parent's internal parameters are passed on to all of its components, however. An example of this is the COLOR parameter for the chair; this parameter value is not inherited by the underframe, but it is inherited by the seat and the back.

The inheritance process essentially serves to connect a lowest level specialty associated with a generic parent and a top level specialty associated with one of its generic components. This connection means that any of the variants of the top level specialty associated with the generic component may be a component of at least one of the variants belonging to the lowest level specialty of the parent. This

---

connection will exist only if the characterization of the top level specialty is identical to that of the lowest level specialty. The lowest level specialty of a red chair and the top level specialty of a red seat can be used to illustrate this. There are four variants of a red chair: a red swivel chair with wheels, a red non-swivelling chair with wheels, a red swivel chair without wheels and a red non-swivelling chair without wheels. Any of the variants of the high-level red seat option may be used as a components for any of the variants of the red chair. The red seat has only one variant in this example, however.

The prerequisite conditions for a high-level component option to be able to inherit parameter values from a parent lowest level specialty are:

1. the external parameters of the component must be a subset of the internal parameters of the parent;
2. the parameter values of the parent lowest level specialty must be a subset of the parameter values of the high-level component option with respect to the external parameters of each of the components.

Several examples are presented in Figure 5.11 to illustrate how these conditions apply. A specific high-level component option is used with different bottom-standard parent options in the first five examples. Parameters identified as p1, p2 en p3 are used in these examples. The parameter value of parameter p1 is equal to v11, v12 or v13. The parameter values assigned to p2 and p3 are v21 and v31, respectively.

The second and fourth examples presented here do not satisfy the two prerequisite conditions for inheriting parameter values. In the second example, the parameter value of v13 that is assigned to parameter p1 associated with the lowest level specialty does not occur for the top level specialty and, thus, does not satisfy the second condition. In the fourth example, parameter p2 associated with the top level specialty does not occur for the lowest level specialty and, thus, does not satisfy either of the two prerequisite conditions.

example	parent lowest level specialty		component top level specialty		satisfies	
	parameter	parameter value	parameter	parameter value	condition 1	condition 2
1	p1,p2	v11,v21	p1,p2	v11,v12,v21	yes	yes
2	p1,p2	v13,v21	p1,p2	v11,v12,v21	yes	no
3	p1,p2	v11,v12,v21	p1,p2	v11,v12,v21	yes	yes
4	p1	v11	p1,p2	v11,v12,v21	no	no
5	p1,p2,p3	v11,v12,v31	p1,p2	v11,v12,v21	yes	yes
6	p1,p2,p3	v11,v12,v31			yes	yes

Figure 5.11: Examples to illustrate inheritance conditions for specialties

The top level specialty shown in the sixth example has no parameters and no parameter values. As a result, this top level specialty can also be seen as a universal top level specialty. The set of external parameters is empty and can also be viewed as a subset of the internal parameters of the parent. This means that this example satisfies the first prerequisite condition. The same reasoning can be followed to reach a similar conclusion with respect to the second prerequisite condition.

### Data model

Various aspects of the data model for a generic bill-of-material will be reviewed in this section using a number of data models. The valid parent/component combinations shown in Figure 5.8 will be used here as the basis for this discussion. A number of data models will be investigated here in the following sequence:

1. generic bills-of-material comprised only of GP's that are defined and identified directly without using parameters;
2. generic bills-of-materials with GP's that are defined and identified directly, regardless of whether parameters are used;
3. generic bill-of-material with GP's that are defined and identified directly, regardless of whether parameters are used, or else indirectly using parameters.

The first type of generic bill-of-material is just a normal bill-of-material as described previously in Chapter 4. In fact, all of the GP's that are identified directly without parameters are comprised of only a single variant; this means that they can be fully identified by the product number. As a result, the data model for this type of generic bill-of-material is essentially the same as the data model for a normal bill-of-material (Figure 4.6). The product and bill-of-material relationship entities only need to have different names: generic product and generic bill-of-material relationship.

This is illustrated in Figure 5.12 where the construction of the seat in the example of the chair is shown. The seat is comprised of a seat frame and 0.5 m<sup>2</sup> of upholstery. If only a single variant of the seat is available, then the generic bill-of-material includes sufficient information to describe a seat fully. The product code of 7419G is sufficient to identify and specify the seat.

In the case of the second type of generic bill-of-material it is also permissible to use GP's that can be specified directly with parameters. The data model used for the first type of generic bill-of-material must be extended in this case with three new entity types:

- a parameter entity type with key = parameter number;
- a generic product/parameter relationship entity type with key = generic product number, parameter number;
- a parameter value entity type with key = parameter number, parameter value.

entity type: GENERIC PRODUCT		
product code	unit	description
3265G	piece	seat frame
7419G	piece	seat
8113G	m2	upholstery

entity type: GENERIC Bill-of-material RELATIONSHIP			
product code of parent	sequence number	product code of component	bill-of-material quantity
7419G	1	3265G	1
7419G	2	8113G	0.5

seat            seat frame  
upholstery

Figure 5.12: Example of the first type of generic bill-of-material

To illustrate this, we can extend the example presented in Figure 5.12. The seat product has two different variants: a red variant and a blue variant. The variants are based upon the choice of upholstery used to upholster the seat. The construction of both seat variants is otherwise identical. As a result, the specification of the entities in Figure 5.12 remains unchanged. Parameter descriptions are used to distinguish between the two different seat variants. Three new entity types need to be specified in order to provide for these parameter descriptions (see Figure 5.13).

To start with, the COLOR parameter must be defined in the parameter entity type. Then the parameter values of RED and BLUE need to be defined for the COLOR parameter in the parameter value entity type. Finally, the relationship between the seat GP and the COLOR parameter needs to be defined in the generic product/parameter relationship entity type.

entity type: PARAMETER		
parameter number	description	
92	COLOR	

entity type: PARAMETER VALUE		
parameter number	parameter value	
92	RED	COLOR
92	BLUE	

entity type: GENERIC PRODUCT/ PARAMETER RELATIONSHIP		
product code	generic product	parameter number
7419G		92
8113G		92

Figure 5.13: Example of the first extension of the first type of generic bill-of-material

It is not possible to exclude the use of certain combinations of parameter values with this data model. Furthermore, it is not possible to record information for a specific variant based upon this data model. An example of this could be planning data for a specific variant. A specialty entity type can be created to resolve these shortcomings. Each variant can then be defined as a specialty in this new entity type and each of the variants can be characterized by one or more parameter values. The relationships between variants and parameter values can be defined in this specialty/parameter value relationship entity type. This enhancement is illustrated in Figure 5.14 for our seat example. This means that the data model has been extended with the following entities:

- a specialty entity type with key = generic product number, specialty number;

- a specialty/parameter value relationship entity type with key = generic product number, specialty number, parameter number, parameter value.

entity type: SPECIALTY	
product code of the generic product	specialty number
7419G	1
7419G	3
8113G	1
8113G	3

entity type: SPECIALITY/PARAMETER VALUE RELATIONSHIP			
product code of the generic product	specialty number	parameter number	parameter value
7419G	1	92	RED
7419G	3	92	BLUE
8113G	1	92	RED
8113G	3	92	BLUE

Figure 5.14: Example of the second type of generic bill-of-material

Note that a distinction between a top level specialty and a lowest level specialty is not yet required at this point since the decision trees have not yet been defined in the data structure.

Since each variant corresponds to a single specialty, variant-specific data can now be recorded. Furthermore, each specialty can be identified using parameter values. To support this, all of the permissible combinations of parameter values should also be defined. The key for a given specialty is defined as the generic product number and a randomly chosen specialty number.



---

The data structure for the second type of generic bill-of-material is presented in Figure 5.16 (within the dotted lines).

The third type of generic bill-of-material provides for specifying a GP indirectly using parameters. This means that the data structure needs to be extended to allow for a number of possible choices. This can be accomplished using decision trees. (Other ways of doing this will be described later.) A choice relationship entity type with a key (comprised of generic product number, sub-specialty and super-specialty) has been defined for this purpose. Two relationships can be identified between the specialty entity type and the choice relationship entity type: the explosion relationship and the implosion relationship (see Figure 5.12). It is possible to determine whether a specialty is a top level specialty or a lowest level specialty by referring to these two types of relationships. This is because a top level specialty will have no implosion relationships and, similarly, a lowest level specialty will have no explosion relationships.

A universal top level specialty can be identified from the fact that it will have no parameter values. This means that there are no specialty/parameter relationship entities defined for this type of specialty.

The third type of generic bill-of-material can be illustrated using the example of the chair underframe (Figure 5.10). The relevant entity types are identified in Figure 5.15 and defined for the example of the chair underframe.

The complete data structure for the generic bill-of-material is presented in Figure 5.16.

The data base can be used to easily identify and specify a given GP. To start with, we will show how this is done for the generic primary products (GPP's) and then for the generic subassemblies (GS's).

If a given GPP does not have any parameters, then it can be specified directly without parameters. If the specialties associated with a given GPP are defined using parameter values but without decision tree relationships, then the GPP can be specified directly using parameters. A GPP is specified indirectly using parameter

values when the specialties of the GPP are defined based upon parameter values as well as decision tree relationships.

entity type: GENERIC PRODUCT		
product code	unit	description
6439G	piece	wheel
7291G	piece	support
7453G	piece	underframe

entity type: GENERIC Bill-of-material RELATIONSHIP			
product code of the parent	sequence number	product code of the component	quantity-per
7453G	1	7291G	1
7453G	2	6439G	5

underframe support  
wheel

entity type: PARAMETER	
parameter number	description
91	WHEELS
93	SWIVEL

entity type: PARAMETER VALUE	
parameter number	parameter value
91	YES
91	NO
93	YES
93	NO

WHEELS

SWIVEL

entity type: GENERIC PRODUCT / PARAMETER RELATIONSHIP	
product code GP	parameter number
6439G	91
7291G	91
7291G	93

wheel

support

support

WHEELS

WHEELS

SWIVEL

entity type: SPECIALTY	
product code generic product	specialty number
6439G	1
7291G	1
7291G	2
7291G	3
7291G	4
7291G	5
7291G	6
7453G	1
7453G	2
7453G	3

wheel

support

underframe

WHEELS=YES

WHEELS=NO, SWIVEL=YES

WHEELS=NO, SWIVEL=NO

WHEELS=YES, SWIVEL=YES

WHEELS=YES, SWIVEL=NO

WHEELS=YES

WHEELS=NO

WHEELS=YES

WHEELS=NO

entity type: CHOICE RELATIONSHIP		
product code generic product	sub- specialty	super- specialty
7291G	5	3
7291G	5	4
7291G	6	1
7291G	6	2
7453G	3	1
7453G	3	2

entity type: SPECIALTY/PARAMETER VALUE RELATIONSHIP			
product code GP	specialty number	parameter number	parameter value
6439G	1	91	YES
7291G	1	91	NO
7291G	1	93	YES
7291G	2	91	NO
7291G	2	93	NO
7291G	3	91	YES
7291G	3	93	YES
7291G	4	91	YES
7291G	4	93	NO
7291G	5	91	YES
7291G	6	91	NO
7453G	1	91	YES
7453G	2	91	NO

wheel	WHEELS
support	WHEELS
	SWIVEL
	WHEELS
	SWIVEL
	WHEELS
	SWIVEL
	WHEELS
	SWIVEL
	WHEELS
	WHEELS
underframe	WHEELS
	WHEELS

Figure 5.15: Example of the third type of generic bill-of-material

The identification of a given GS can be done directly, without parameters when the GS does not have any parameters and, furthermore, all of its components can also be specified directly, without parameters. A given GS is specified directly, with parameters, when the specialties of the GS have parameter values but do not have

choice relationships and, furthermore, all of its components can be specified directly. A given GS is specified indirectly when it cannot be specified directly.

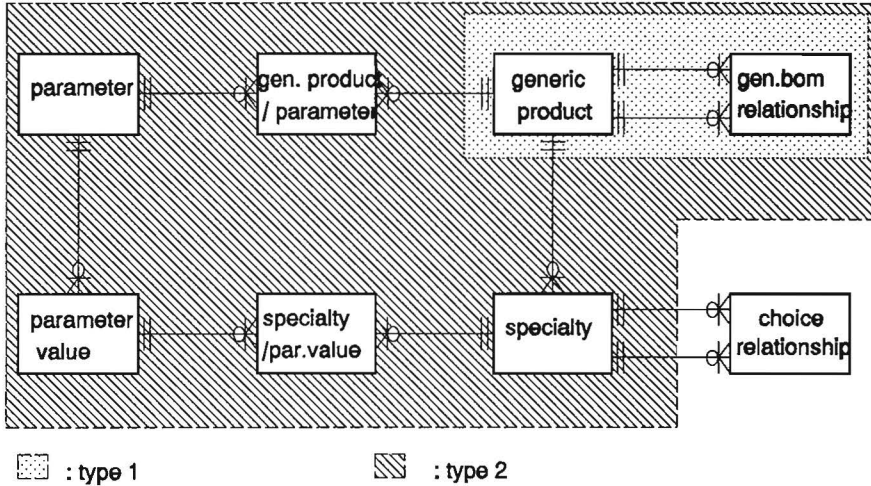


Figure 5.16: Data structure for a generic bill-of-materials system

### Summary

The concept of a generic bill-of-material was developed in this chapter. After identifying the types of variants that can be specified within the existing bill-of-material systems for product families, we have demonstrated that three types of identifications can be used within a generic bill-of-material. These three types of identifications are also evident from the differences in the data structure associated with the generic bill-of-material. One type of limitation has been discussed in this chapter with respect to the choice of parameter values, namely, a situation in which the same choice of parameter values needs to be made for the components of a product. In the next chapter, we will explain how other limitations can be recorded in the generic bill-of-material.

### **Introduction**

One type of decision situation has already been discussed in the previous chapter. This involved the creation of an environment to enable the coordination of the choice of generic primary products (GPP's). This chapter deals with several different methods for recording restrictions other than the coordination of GPP's.

### **Decision trees**

As discussed in the previous chapter, a variant of a finished product in a generic bill-of-material is determined by combining GPP variants. More specifically, the variants of the GPP's are used that are found at the beginning of the paths leading to the finished product. Some of the combinations of GPP variants may be invalid, however. This is explained below for the generic products that can be specified directly and indirectly.

#### Direct identification

The valid combinations of parameter values can be recorded by listing the valid variants in the case of GP's that can be specified directly with parameters. These variants can be seen as the specialties of the GP. These specialties may be specialties at the top-as well as at the lowest level. It is important to note that a decision tree cannot be used in this situation. The variants of the associated GPP's are also determined through the inheritance process. These GPP's fall into the direct identification category, either with or without parameters.

To illustrate this, consider the generic seat product with the red and blue variants. These variants can be seen as the two specialties of the seat product. The inheritance process then ensures that the red upholstery will always be chosen for the red seat variant.

#### Indirect identification

The valid combinations of GPP variants can be recorded by using decision trees in the case of GP's that can be specified indirectly with parameters. These decision trees may be used in connection with the finished product, itself, and also in

connection with its direct and indirect components. As previously seen, the decision trees may only be used with GP's that are specified indirectly with parameters.

A decision tree associated with a GP can be seen as a acyclical, directed graph. The nodes of this graph are the specialties of the GP. The branches of this graph are the choice conditions and relationships of the decision tree. A sub-specialty is represented by the node at the end of a branch and a super-specialty is represented by the node at the beginning of a branch. The definition of branches in this way means that a given sub-specialty is a subset of the associated super-specialty. The graph theory term for a lowest level specialty is a "leaf," while a top level specialty is referred to as a "root."

One of the sub-specialties must be chosen for each of the specialties when the decision tree is followed. Since a specialty is actually a set of variants, a sub-specialty can be seen as subset of this set of variants. The choice becomes more specific as the decision tree is followed. The set of valid variants for a given product becomes usually increasingly smaller. This means that there are fewer variants per specialty as one follows a identification path leading from a top level specialty to a lowest level specialty.

The set of sub-specialty variants is not necessarily smaller than the set associated with the super-specialty for a given branch of the decision tree. The number of variants may also be the same, making the two sets identical. The only difference between the sets in this case is how the sets are defined. A sub-specialty will always have at least one parameter that is different from the parameters of the associated super-specialty. Essentially, a **translation** has taken place to convert the super-specialty parameters to the sub-specialty parameters.

A super-specialty with its associated branches and sub-specialties can also be viewed as a production rule in an expert system. This type of production rule consists of two parts: a premise and an action.

---

A production rule can, thus, be formulated as follows:

If

the variant (V) to be specified belongs to the super-specialty P

then

variant V will belong to one of the sub-specialties of P.

Since specialties are defined in terms of parameter values, the aforementioned production rules can also be formulated in a different way as follows: if the variant (V) to be specified can be defined by the parameter values of the super-specialty P, then V can be defined by the parameter values of a sub-specialty belonging to P.

This can be illustrated using our chair example. We need to extend the example (as presented in terms of a data model in Chapter 5) as follows: the color GRAY is added as an option and the chair can be supplied with or without armrests. This is accomplished by defining an ARMREST parameter with the possible values of WITH ARMREST and WITHOUT ARMREST. In addition, two restrictions are to be defined with respect to the valid chair variants:

1. all of the variants with wheels must necessarily include the swivel option;
2. all of the blue variants are deluxe models that are, by definition, always supplied with wheels.

The valid combinations of the relevant parameter values associated with these restrictions are presented in Figure 6.1.



WHEELS	SWIVEL	C O L O R		
		red	gray	blue
yes	yes	yes	yes	yes
	no	no	no	no
no	yes	yes	yes	no
	no	yes	yes	no

Figure 6.1: Possible combinations of parameter values for the WHEELS, SWIVEL, and COLOR parameters

### Condition GP

Before we can record the aforementioned restrictions and conditions in the form of a decision tree, it must be determined which GP should be used for documenting these restrictions and conditions. As previously seen, there are always two sets of variants associated with a condition. One set is related to the premise of the condition and the other set is related to the action associated with the condition. These two sets need to be defined as specialties of a single GP, referred to as the "condition GP." The set associated with the premise is the super-specialty of a branch in the decision tree. The set associated with the action is the sub-specialty of the same branch. When a choice needs to be made in the action associated with the condition, then multiple sub-specialties need to be defined for the super-specialty involved. This means that multiple branches will also exist in this case.

The super-specialty and the sub-specialty are characterized by parameter values. We can assume that these specialties are specialties of a GP called "Y," where Y is the condition GP that has a super-specialty called "SP." If we use SP as the basis for making a specific choice "K" from the available sub-specialties, then the parameter values for SP are, by definition, already known. There are two possibilities for documenting these parameter values associated with super-specialty SP in the generic bill-of-material. The first possibility is to record the parameter values as part of the generic product Y associated with super-specialty SP. This is at a higher

---

level in the decision tree associated with super-specialty SP. A requirement here is that the parameter values be recorded along with a path that leads to SP. The other possibility is to record the parameter values as part of a different GP, GP "X" at a higher level (i.e., with a lower lowest level code) in the generic bill-of-material path leading to GP Y with super-specialty SP.

It is always possible to determine the parameter value of a super-specialty at the level of a GP X, provided that the following conditions are satisfied:

- .1 GP X must have common path with the condition GP (GP Y)
- .2 GP X must be at the same level as, or higher than, GP Y in the generic bill-of-material.

If GP X is higher than GP Y in the generic bill-of-material, then the parameter value needed by the super-specialty will be transferred through any GP's which may lie on the relevant path. This means that these intermediate GP's, if any, must have specialties which are also characterized by the parameter of the super-specialty. At least one of the top level specialties and one of the lowest level specialties of these GP's must contain the relevant parameter value needed by the sub-specialty SP.

The question must then be answered of where the condition GP, GP Y, should be located in the generic bill-of-material. In other words, which GP should be used as GP Y. We can define two further aspects:

- . GP X is the GP where the parameter values for the premise of the condition are chosen
- . GP Z is the GP where the parameter values for the action associated with the condition are used (and where these parameter values were chosen prior to implementing the condition).

The condition GP (GP Y) should be the direct or indirect parent of the GP where the parameter values for the sub-specialty were determined before implementing the condition (GP Z).

Therefore, GP Y should lie on the path between GP Z and GP X. If GP Z is a component of multiple GP's with the same condition, then a variety of paths are suitable for establishing the link with GP Y. GP Y should then lie on a common

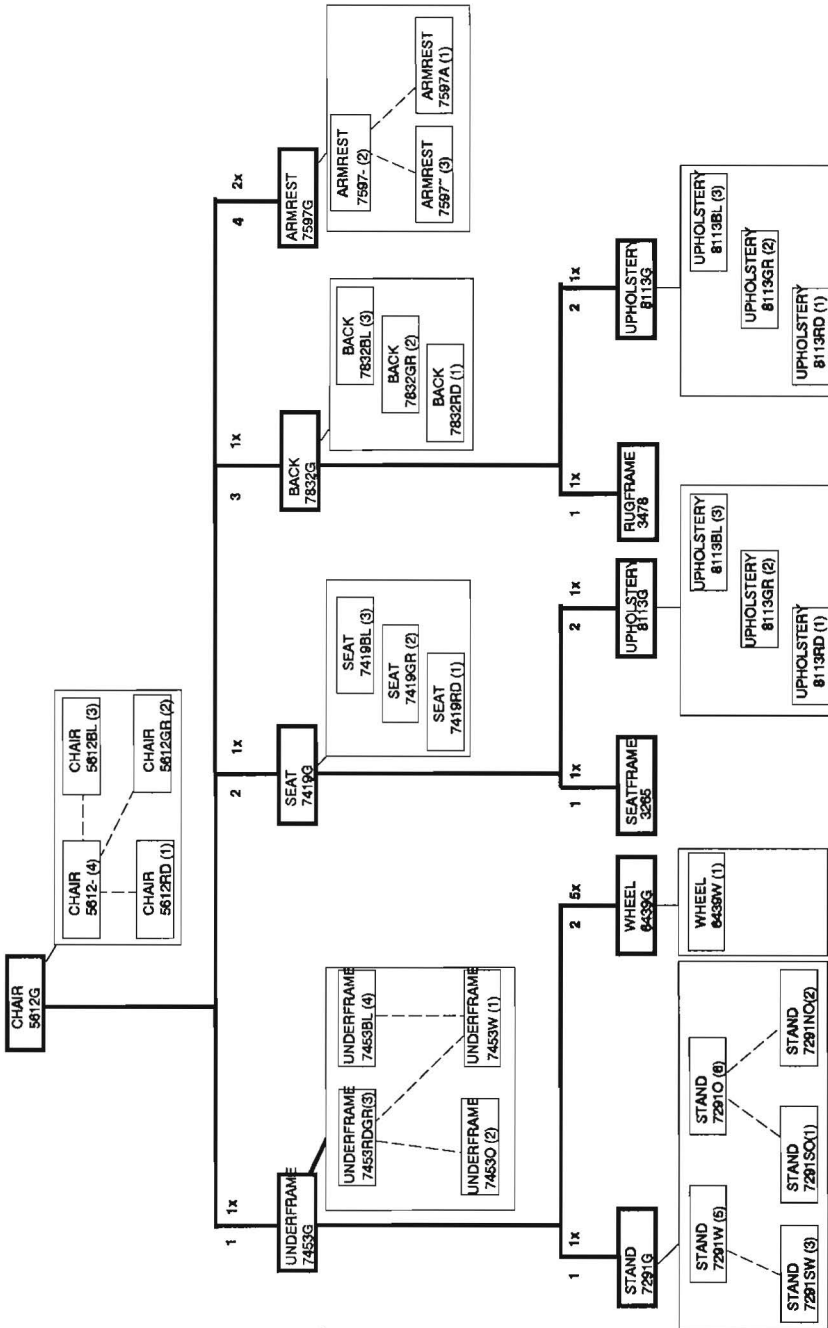


Figure 6.2: Complete generic bill-of-material for the chair example

---

segment of these paths. Since the intention is to keep the parameter choice at the lowest possible level in the generic bill-of-material (as we have seen in the previous chapter), GP Y should coincide with GP Z. This means that the condition should be implemented at the GP where the sub-specialty parameter was originally determined.

This line of thought can be applied to the two conditions that we wish to implement in the chair example.

The first condition was that all of the variants of the chair with wheels should be swivel chairs. The premise for this condition is incorporated within a super-specialty characterized by the parameter value of YES for the WHEELS parameter. The super-specialty's parameter value of YES for the WHEELS parameter is dictated by the underframe GP. This is our GP X. The action of the condition is represented by a sub-specialty characterized by the parameter value YES for the SWIVEL parameter in addition to the super-specialty's parameter value. The SWIVEL parameter value was originally chosen at the stand GP. This means that the stand is our GP Z. As explained above, the condition should be placed at GP Z, or the stand GP in this case. The super-specialty's parameter should be inherited from the underframe. This had already occurred in our original example. Implementing the condition has only led to a situation in which the branch has been eliminated that links the top level specialty of the stands with wheels and the lowest level specialty of the non-swivel stands with wheels.

The second condition was that all of the blue variants should have wheels. The super-specialty for this second condition consists of the blue variants. The sub-specialty consists of the variants with wheels. The super-specialty's parameter value of BLUE for the COLOR parameter is dictated by the chair GP. Since the sub-specialty's parameter value is dictated by the underframe GP, the condition will need to be added as a branch in the decision tree. A top level specialty characterized by the color BLUE will need to be added to the underframe. The branch represents the relationship between this top level specialty and the lowest level specialty of the variants with wheels. This second condition implies, indirectly, that the variants of any color other than BLUE may occur in a variant with wheels as well as a variant without wheels. This means that a top level specialty with the gray and red variants will need to be added. Two branches will

---

connect this top level specialty to the lowest level specialties for the underframes with wheels and without wheels. The complete generic bill-of-material for this chair example is presented in Figure 6.2.

### **Complete decision tree for a GPP**

A complete decision tree can be constructed based upon the generic bill-of-material. This includes all of the possible choices that are relevant for defining each GPP variant at the beginning of a path leading to the finished product to be specified. This complete decision tree can be constructed based upon the decision trees of the GP's lying on the path between the relevant GPP and the generic finished product.

This can be illustrated using the same example. Assume that the GPP is the stand in this case. The path leading from the stand to the generic finished product of the chair includes the following generic products: a stand, an underframe and a chair. Each of these GP's has one or more decision trees associated with it. These decision trees are then integrated with each other based upon the inheritance mechanism. The complete decision tree for the stand is shown in Figure 6.3. The solid branches represent the connections created by the inheritance mechanism while the dotted lines represent the branches from the original decision trees of the GP's.

It is not always necessary to include all of the GP's that lie on the path to the finished product in order to complete the decision tree. This situation occurs when a universal top level specialty is present in one of the decision trees on the path. If we assume that this top level specialty is a specialty of GP X, then the decision trees for the GP's lying on the path between GP X and the finished product will not be required for completing the decision tree. This is because a universal top level specialty is neutral within its environment. This means that this top level specialty is always the last step in the decision process for a GPP, regardless of which choices have been made along the way. The GPP is also independent of the choices made in the decision trees associated with the GP's between GP X and the finished product. An example of this is the complete decision tree for the armrest GPP. This is easy to see from Figure 6.2. The complete decision tree for the

armrest is identical to the decision tree for the armrest by itself since the armrest contains a universal top level specialty.

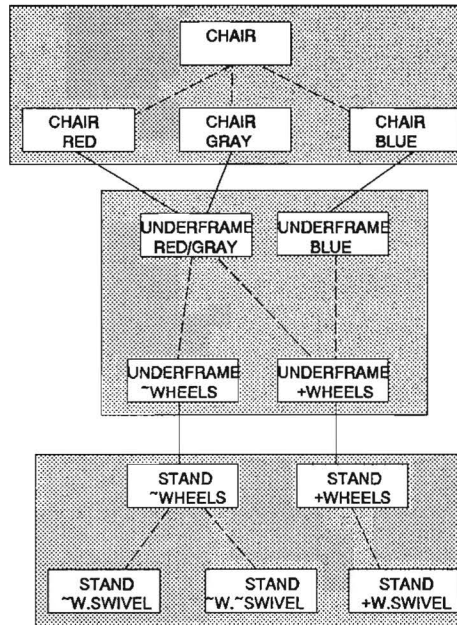


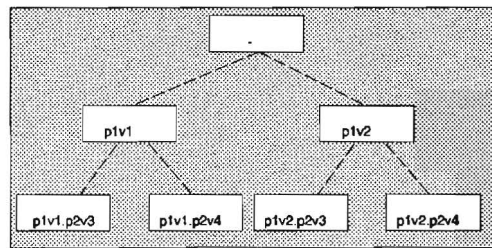
Figure 6.3: Complete decision tree for the vertical support

It is also possible to produce a complete decision tree for the finished product. A complete finished product decision tree is the combination of all of the complete GPP decision trees. This tree structure can be drawn easily for our chair example by adding all of the inheritance paths (see Figure 6.2.).

### Evaluating the decision tree

The generic bill-of-material concept decomposes the choices and the restrictions. They are assigned to the generic products in accordance with the principles of GPP coordination and condition assignment. There are no rules to be given for constructing decision trees other than the guidelines already presented for implementing conditions. The sequence of decision-making in this respect is random. This means that there is no unique design for a decision tree.

One problem with decision trees is the redundancy of data, especially with the larger decision trees. In the example presented in Figure 6.4, for example, a top level specialty is divided into four lowest level specialties. Each lowest level specialty is characterized by parameters  $p_1$  and  $p_2$ . Each parameter has two possible values:  $v_1$  and  $v_2$  for  $p_1$ ,  $v_3$  and  $v_4$  for  $p_2$ . This means that two decisions need to be made in order to arrive at one of the lowest level specialties, one decision to choose a value for parameter  $p_1$  and a second decision to choose a value for parameter  $p_2$ .



$p_{jvk}$ : parameter value  $k$  for parameter  $j$

Figure 6.4: Decision tree

### Decomposing the decision tree

We can divide the decision tree into similarly-structured sub-trees in order to reduce the amount of data that needs to be recorded in the decision tree. The sub-trees starting at  $p_{1v1}$  and  $p_{1v2}$  in the example of Figure 6.4 are similarly-structured. Similar details appear in both sub-trees. Using these similarities, the example presented in Figure 6.4 can be translated into the structure presented in Figure 6.5. Two decision trees have been created in addition to the top level specialty and the lowest level specialties. A specialty is detailed in each of the decision trees by identifying two specialties characterized by a parameter value. The super-specialty is a specialty characterized by the complete set of parameter values for a given parameter. This approach saves two branches but, on the other hand, creates the need for defining two additional specialties.

---

The savings to be gained are more evident in the case of larger decision trees. If we add four extra levels to the decision tree illustrated in Figure 6.4 and each decision level provides a choice between two alternatives, then this will result in a total of 64 lowest level specialties at the sixth and lowest level in the tree. A total of 62 specialties will be required at the intermediate levels. There will be 126 branches (one branch for each non-top level specialty). On the other hand, this decision tree could be replaced by six single-level decision trees to be used in parallel. The number of intermediate specialties would be reduced from 62 to 18 and the number of branches reduced from 126 to 12 in this case.

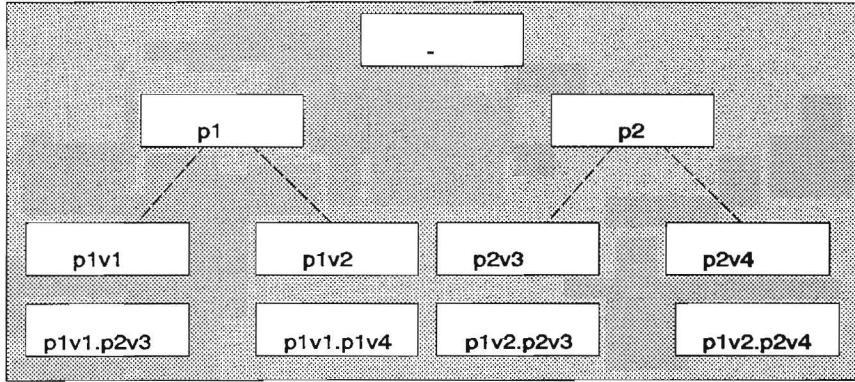
The data model presented in Chapter 5 requires only minor changes to reflect this difference. Two attributes will need to be added to each specialty to indicate whether it is a top level specialty and to indicate whether it is a lowest level specialty.

A further reduction of data is possible in some cases. This is possible when all of the values associated with a parameter are always valid within a GP. The valid values are already recorded within one of the entity types in this case. This means that it is not necessary to record these values again using a single-level decision tree. We will not make use of this form of data reduction here since we eventually wish to add additional data to the choices recorded at the various branches.

Conditions will be recorded in decision trees consisting of one or more levels. The super-specialty at the top of such a decision tree is always characterized by one or more parameter values.

In the case of a generic bill-of-material, it is necessary to ensure that only a limited number of lowest level specialties are offered when a finished product is specified based upon a top level specialty. This means that, minimally, the top level specialties and lowest level specialties will need to be identified for each GP. In addition, it is necessary to know which combinations of top level specialties and lowest level specialties are valid.





pjvk: parameter value k for parameter j

Figure 6.5: An alternative decision tree

### Decision table

Other methods can be used in addition to decision trees to record the valid combinations. One of these other methods is the use of decision tables [CODA83]. A **decision table** is based upon decision rules. Each **decision rule** consists of a number of conditions and a number of actions. These actions are performed when all of the conditions associated with the rule have been met. Each decision table contains a group of related decision rules. The most important aspects of a decision table are illustrated in Figure 6.6. The top left quadrant contains a description of the conditions while the bottom left quadrant contains a description of the possible actions. The condition entries and the action entries are found in the top right, respectively, bottom right quadrants. Each column for the condition entries and action entries represents a decision rule.

We first need to define the conditions and the actions involved in specifying a lowest level specialty in order to be able to use a decision table to record the valid combinations within a generic bill-of-material. The conditions in the decision table refer to the valid parameter values or the valid combinations of parameter values.

The actions in the decision table represent the possible choices of a lowest level specialty. This can be illustrated using the conditions associated with the underframe (Figure 6.7).

	decision rule						
	1	2	3	4	5	6	
condition description	condition entries						
condition 1	y	y	y	n	n	n	y: yes, the condition is true
condition 2	y	n	y	n	y	n	n: no, the condition is false
condition 3	y	n	n	-	y	-	-: n.a., the condition is not applicable
action description	action entries						
action 1	x	-	x	-	x	-	x: action
action 2	x	x	-	x	x	x	-: no action

Figure 6.6: Decision table

This type of decision table uses only a limited number of the possible condition entries and action entries. For this reason, we call this a decision table with **limited entries**. On the other hand, if we do not limit the condition entries to the values of "yes", "no" and "not applicable" and do not limit the action entries to "action" and "no action", then we would refer to this as a decision table with **extended entries**.

In the decision table for the underframe, the condition entries are the valid parameter values. The decision table presented in figure 6.7 can be expressed in the form of the table shown in figure 6.8 in this case. The action entries for this table are the numbers 1 and 2, representing the lowest level specialty with wheels and without wheels, respectively.

	1	2	3	4	5
COLOR = RED	y	y	n	n	n
COLOR = GRAY	n	n	y	y	n
COLOR = BLUE	n	n	n	n	y
WHEELS = YES	y	n	y	n	y
WHEELS = NO	n	y	n	y	n
choose l.l.sp. 7453-1	x	-	x	-	x
choose l.l.sp. 7453-2	-	x	-	x	-

Figure 6.7: Decision table for the GP underframe

	1	2	3	4	5
COLOR	RED	RED	GRAY	GRAY	BLUE
WHEELS	YES	NO	YES	NO	YES
choose l.l.sp. 7453-	1	2	1	2	1

Figure 6.8: Revised decision table for the GP underframe

Two requirements are important in this situation to ensure proper decision tables. The first requirement is that a decision table must be **complete**. A complete decision table is one in which there are no missing decision rules. In other words, actions must be defined for every possible condition. The decision table in Figure 6.8 is not complete since the combination of the conditions blue and without wheels is missing. If we add a production rule to cover this combination of conditions, then the table is complete. The associated action is "no action" (or "-" in the table) in this case.

The second requirement for a proper decision table is **consistency**. This means that there may be no invalid combinations of conditions in the table and that there are no actions missing for any of the decision rules in the table. The use of a generic bill-of-material approach ensures that this last requirement is satisfied. The action resulting from each decision rule is always a choice of whether to choose a specific lowest level specialty.

**Multi-dimensional decision table**

Since only one action may appear in the decision table with a generic bill-of-material and the decision table is complete, this type of decision table can also be represented in the form of a multi-dimensional matrix. A parameter then represents a dimension of this matrix and the action input becomes a matrix element. The action entries is always the choice of either selecting a lowest level specialty or "no action." A matrix element is characterized by parameter values. This type of matrix is referred to as a **multi-dimensional decision table** in the remainder of this thesis.

This can be demonstrated using the example above. The WHEELS and COLOR parameters represent the two dimensions of the matrix. The elements or actions to be performed are choosing the lowest level specialties. This multi-dimensional matrix is shown in Figure 6.9.

underframe lowest level specialty 7453-		WHEELS	
		YES	NO
COLOR	RED	1	2
	GRAY	1	2
	BLUE	1	-

Figure 6.9: Multi-dimensional decision table for the GP underframe

The condition entries may be a combination of parameter values instead of just a single parameter value.

The major advantage of using such a decision table is its ease-of-use. A decision table is particularly useful for presenting the valid combinations of parameter values. It is less convenient to use a decision table for documenting the conditions, however, since all of the top level specialties and lowest level specialties must be recorded along with their respective parameter values. The top level specialties and lowest level specialties must be recorded to support the inheritance mechanism. In

addition, a variety of transaction and planning data needs to be recorded with the lowest level specialties. This is discussed in more detail later. There is not much additional information required in order to identify all of the possible choices. If, in spite of these shortcomings, decision tables are used to document the possible choices, then a certain amount of data redundancy will be present.

A decision table could be used as a "user interface" for the configurator. The required decision table could be generated directly from the decision tree.

### **Production rules**

Another way to record conditions is to make use of **production rules**. As we have seen previously, a production rule consists of a **premise** and an **action**. A production rule is equivalent to a decision rule in a decision table. The premise corresponds to the condition entries of a decision rule. the action is obviously similar to the action entries of a decision rule. We have already been able to draw the conclusion that a single-level decision tree is equivalent to a production rule. This suggests that the use of production rules to record data will not provide any additional advantages over the use of a decision tree. Therefore, we will only make use of decomposed decision trees for recording conditions in the remainder of this thesis.

### **Continuous parameter values**

Until now, we have assumed that a parameter will only have a limited number of parameter values. This may not always be the case in practice, however. One could think of a variety of measurements for products, such as the length, width and height. We will make allowances here for the use of continuous parameters in order to ensure the applicability of generic bills-of-material in a wide range of situations. This, of course, leads to a number of consequences with respect to the generic bill-of-material.

One consequence concerns the identification of a variant. Three types of identification were discussed in Chapter 5: directly without parameters, directly

with parameters and indirectly with parameters. The use of continuous parameter values will, of course, only affect the last two types of identification.

The specialty number of the variant is no longer sufficient in the case of identifying a variant directly with parameters when one or more of the parameters has a continuous parameter value. This is due to the fact that a lowest level specialty belonging to this type of GP will have either one or no variants. This means that the variant was identified simply by choosing this specialty. This specialty may have a large number of variants when a parameter with a continuous value is used. All of the continuous parameters will need to have a specific value before a specific variant can be selected.

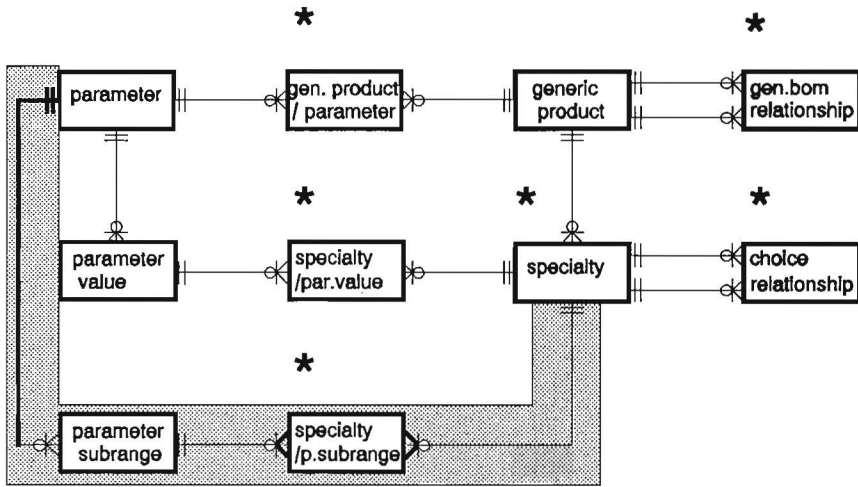
This can be illustrated using the following example. Assume that synthetic construction board measuring 150 x 40 cm. can be ordered in two colors: white and brown. Versions of this material that are cut to a specific size can be considered to be variants of the generic product called synthetic construction board. The two colors of construction board can be considered to be specialties of this GP. If we wish to specify a brown piece measuring 100 x 20 cm., then the desired color as well as the desired size must be provided. In other words, the full identification would be:

·	the generic product:	3544G synthetic construction board
·	the specialty:	2 brown, 150 x 40 cm
·	the continuous "length" parameter:	100 cm
·	the continuous "width" parameter:	20 cm

A second consequence of using continuous parameters is the fact that we cannot record all of the possible parameter values, individually. It must be possible to specify a series or range of values for this type of parameter. This type of range is referred to here as a **parameter subrange**. One or more parameter subranges may be specified for any given continuous parameter. Each parameter subrange is defined by a beginning value, an ending value and a step size. The step size is not strictly necessary for defining a subrange, however, this is useful for specifying the desired accuracy for measurements. The following example illustrates the

usefulness of parameter subranges. Assume that a certain length may vary between 1 and 2 meters with a step size of 1 cm. this means that this length parameter can have 101 possible values. These values could be recorded as 101 individual values or otherwise recorded in the form of a parameter subrange. Significantly fewer data elements are required to record these 101 values. The disadvantage of using a parameter subrange is that transaction data and planning data cannot be associated with just a single value.

A specialty can now be described in terms of parameter values or parameter subranges. This affects the data structure of the generic bill-of-material. To start with, a PARAMETER Subrange entity type will need to be added. An entity type called SPECIALTY/PARAMETER Subrange will also be required. This leads to the data structure diagrammed in Figure 6.10.



\*: relevant to engineering changes

Figure 6.10: Data structure for the generic bill-of-material with continuous parameters

---

Finally, the use of continuous parameters has implications for the inheritance mechanism. The inheritance mechanism not only serves to connect the parent lowest level specialty and the component top level specialty, but also serves as a means of transferring the chosen parameter values for the continuous parameters from the lowest level specialty to the top level specialty. Furthermore, the parameter values of continuous parameters also need to be transferred from the super-specialty to a sub-specialty in the decision tree.

The use of continuous parameters for variants with indirect identification involves the same consequences as described here for the direct identification.

### **Algorithms**

A decision made by traversing a decision tree can also be made through the use of an algorithm. This type of algorithm is assigned to a super-specialty. Based upon the parameter values of the super-specialty, the algorithm determines the parameter values to be passed on to the sub-specialty. When all of the sub-specialty parameter values are known, the choice of sub-specialty can then be made automatically. If this is not the case, then the total number of possible choices will, nevertheless, be reduced.

An algorithm may be local or global. A global algorithm may be used by multiple super-specialties. A local algorithm may be used by only a single super-specialty. Since a global algorithm can be used by more than one super-specialty and each of these super-specialties may use different parameters in conjunction with different sub-specialties, it is necessary to keep track of which parameter belongs to which algorithm variable. The choice has been made here to use global algorithms in conjunction with our generic bill-of-material in order to prevent data redundancy.

The use of algorithms can be illustrated through the example of our synthetic construction board. Assume that we wish to construct a box using this material (see Figure 6.11).

The construction board is available in various thicknesses ranging from 12 through 20 millimeters. The parameters of the box are the length, the width, the height and



the thickness of the board to be used. In order to determine the dimensions of pieces of board to be cut to make this box, the thickness of the board must be taken into account. For example, the width of the piece needed for the front and back of the box must be equal to the height of the box less the thickness of the board. The generic bill-of-material for the box is presented in Figure 6.12.

In order to use the algorithms in connection with the generic bill-of-material, the data structure will need to be extended somewhat. Three new entity types will be required. To start with, an ALGORITHM entity type will need to be added with the attributes of algorithm identification and description. A STATEMENT entity type will also be needed with the attributes of algorithm identification, statement number and statement contents. Finally, a SUPER-SPECIALTY/ALGORITHM entity type will be required with the attributes of GP code, super-specialty number, algorithm identification, algorithm variable name, status (input or output) and parameter number.

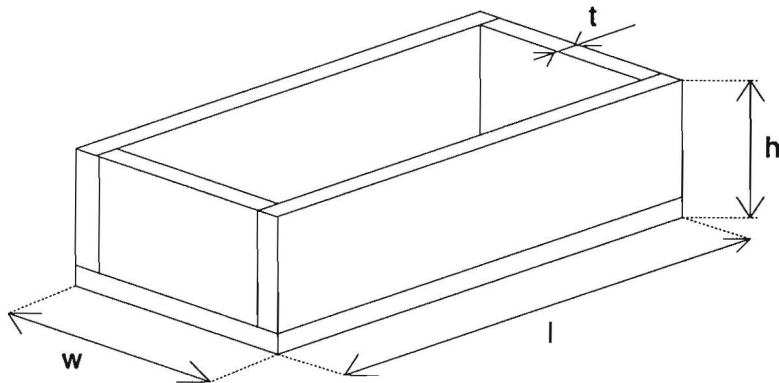


Figure 6.11: Box constructed from synthetic construction board

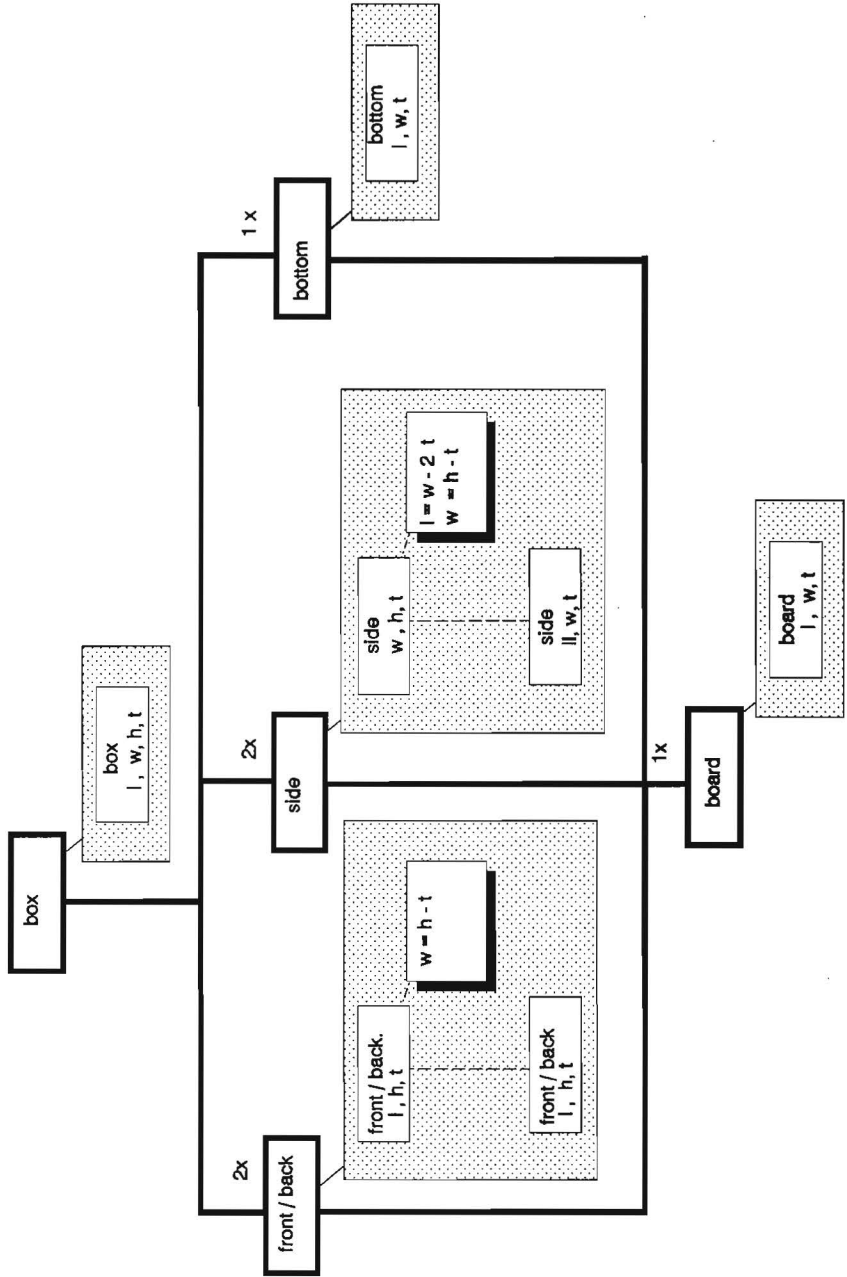


Figure 6.12: Generic bill-of-material for the box

## Engineering change

As seen in Chapter 4, an engineering change can normally be recorded in the bill-of-material when a normal bill-of-material is used. This can be accomplished by recording a valid from-to period for each bill-of-material relationship. Any changes in the way the product is constructed can be documented in this way. A article does not represent just one single product in the case of a generic bill-of-material, but rather a set of products. This has implications for the way in which engineering changes are documented. A change may affect the way in which a product is constructed and may also affect the number of possible variants within a generic product or a specialty. Furthermore, a change may also affect the description of a product. The description is expressed in terms of the parameters and the associated values.

Similar to the way in which changes are recorded in normal bills-of-material, the changes in the generic bills-of-material are recorded in the entity types that refer to relationships. The validity period is similarly associated with these relationships. Six types of relationships are relevant here, namely:

- 1 the generic bill-of-material relationship for the change in the product structure
- 2 the choice relationships for the change in possible choices
- 3 the relationship between specialty and parameter value for the change in the description of a specialty
- 4 the relationship between specialty and parameter subrange for the change in the description of a specialty
- 5 the relationship between generic product and parameter for the change in the description of a generic product
- 6 the inheritance relationship for the change in the product structure and the component choice.

---

This last relationship needs to be explained in more detail. This relationship is only established when it is actually used and is, therefore, not recorded in an entity type. This relationship is created between a parent lowest level specialty and a component top level specialty. The lowest level specialty or the top level specialty may change when an engineering change is implemented. A change in the lowest level specialty can be recorded in the choice relationship that makes this lowest level specialty a sub-specialty. It is also possible that the change involves a top level specialty and is recorded in the inheritance relationship that is the choice relationship that makes this top level specialty a super-specialty. This approach may be tedious when a large number of choice relationships need to be modified. An alternative approach in this case would be to specify a validity period for the specialties, instead. This approach is followed here.

An asterisk is used in Figure 6.10 to indicate the entity types that may be involved in an engineering change.

A large number of changes may need to be made in multiple entity types in order to fully document an engineering change. The data structure needs to be expanded to provide a facility for tracking all of the detailed changes associated with any given engineering change. The CHANGE ORDER entity type has been added for this purpose.

### **Summary**

A number of methods have been described in this chapter to incorporate conditions in a generic bill-of-material. In addition, the possibility for using continuous parameters in addition to discrete parameters has been introduced. The application of algorithms as part of the decision process was then proposed as a more effective way to use continuous parameters and a way to extend the applicability of generic bills-of-material. Finally, an approach to recording engineering changes in a generic bill-of-material was described.

## 7 Pragmatic aspects of the generic bill-of-material

---

### **Introduction**

To conclude this part of our book, we consider how to specify a generic bill-of-material independently of any specific applications. A bill-of-material for a given product can be specified in several different ways as described in Chapter 4. This is primarily dependent upon how a product is viewed. All of the different disciplines within a company will normally have different ways of viewing a product. An engineer needs to define a product in terms of its functional modules. A logistic assistant will, on the other hand, need to see a product in terms of the components that can be kept in stock. These different points of view become apparent from the choice of subassemblies used to define a product. A different choice of subassemblies, in turn, dictates a different way in which a product is constructed. Since a generic product represents a set of products, a bill-of-material for a generic product can be specified according to different points-of-view in the same way as a normal product.

### **Choosing a generic product**

Choosing the most appropriate generic composition is an important aspect of defining a generic bill-of-material, as it is in defining a normal bill-of-material. Before dealing with the question of which generic composition is best, it is necessary to determine which products are to be covered by a given generic product. By definition, a generic product represents a specific set of real products. As discussed previously in Chapter 5, these products need to be constructed in more or less the same way in order to be defined in some sensible way by a single generic product. In an ideal situation with products that are all constructed in the same way, each of the subassemblies of these products will also be defined by a generic subassembly of the generic product. This also means that the product structures of each of the products belonging to a given generic product all will be the same as the generic product structure.

This ideal situation rarely occurs in real life, however. The variants of a finished product often have product structures that are nearly the same, but not identical. This can be illustrated by considering a common situation in which an optional product feature is available that represents a specific variant of a product. The product structure of one variant then becomes different from the product structure

of another variant in this way. The variant that includes the optional feature will have an additional bill-of-material relationship. Nevertheless, these two variants still have product structures that are almost identical and can logically be defined in terms of the same generic product. The "ARMREST" option for a chair is a good example of this where a bill-of-material relationship is defined between the chair GP and the armrest GP in the generic bill-of-material. The armrest GP then has two variants: the variant with armrests and the variant without armrests. This specification associated with the variant without armrests is "empty."

If a large number of products with differing product structures are all included within the scope of a single GP, this leads to a limited number of bill-of-material relationships and a large number of parameters and conditions. The converse is also true: a GP that covers only a few different products will tend to have a large number of bill-of-material relationships and a limited number of parameters and conditions.

### **Choosing generic primary products**

The primary products are found at the lowest level of a bill-of-material. As a result, the primary products never have a product structure. This means that all of the primary products that are clustered and associated with a single generic product must be similar. In other words, they must have a common set of certain properties. At the same time, other properties will be different so that a distinction can be made between the different primary products within the generic primary product (GPP). These properties will be related to the parameters that need to be defined for the GPP. The values associated with these parameters are then used to identify the specific variants covered by a GPP.

If a large number of primary products are clustered within a single GPP, only a few common properties will exist for these primary products. This means that there will then be a large number of properties that may be different from one primary product to the next; the minimum number of parameter values needed to specify a given primary product will tend to be large.

A relatively small number of parameter values will be needed when only a limited number of primary products are clustered within a single GPP. This implies that

---

a large number of GPP's will need to be defined, however. In this case, a large number of GPP's implies that it will be necessary to define many more generic bill-of-material relationships than would otherwise be necessary for only a limited number of GPP's.

### **Generic product structure**

The choice of a generic product structure is dependent primarily upon which point-of-view is used to construct the product. This is discussed in more detail later in this chapter. A second consideration is the levels that are incorporated in defining the product structure. In other words, it is important to consider which subassemblies are used to build a given product. This will depend upon how often an intermediate product level is actually required. The situation is simplified when a given subassembly can be used as a component in a large number of parent products. The use of subassemblies can become burdensome, however, when many subassemblies are defined and need to be maintained. Therefore, it is advisable to define only a limited number of subassemblies.

### **Parameters associated with generic subassemblies**

The choice of parameters to be used in connection with a given generic subassembly depends upon the parameters of the generic primary products that need to be coordinated. When a parameter is used by multiple GPP's and is assigned a single value for configuring a specific finished product variant, then this parameter value should be assigned at the level of the first common parent of the GPP's involved. The parameter value specified at the level of the common parent is then passed on to the relevant GPP's. This means that any intermediate subassemblies must also include this parameter value as a distinguishing characteristic so that it can be passed on. The parameters inherited from the GPP's are always parameters of specialties at the lowest level of the generic subassembly (GS) or internal parameters of the GS.

A second type of parameter is the external parameter. External parameters are used to characterize the situation for which the GS is used in the generic bill-of-material. The choice of a subassembly variant depends upon the parent lowest level specialty. The external parameters are translated into internal parameters at the GS

level. This means that there are relationships between the parameters of the GS parents and the internal parameters of the relevant GS.

### **Conditions and algorithms**

The aforementioned relationships are defined in the form of conditions and algorithms. Conditions are associated with certain GS's as explained in the previous chapter; there, we arrived at the conclusion that a condition should be associated with the GS which deals with the action part of that condition.

The same line of reasoning can be applied to the allocation of an algorithm. An algorithm should be allocated to the GS associated with the results of that algorithm. These results should be recorded as parameter values.

### **Different points-of-view**

The different organizational functions within a company may have differing points-of-view with respect to a finished product. The various points-of-view can be expressed in terms of different ways in which a product is constructed. The differences will be noticeable particularly in the form of different definitions of subassemblies since the different points-of-view must necessarily be based upon the same finished product and the same primary products needed to produce this finished product. The typical points-of-view for constructing products as found in each of the most important business functions are described in the following sections. It is seen how these points-of-view affect the product structure of the generic bill-of-material.

### **Product development point-of-view**

The Product Development Department is responsible for designing the finished product to be delivered to the customer. A developer normally defines the product to be designed in terms of several distinct functional modules. Each function performs a specific task. A set of components is assigned to each function. A drawing is normally produced for each function. When a certain function is already known and has already been tested, previously existing drawings may be used.



---

The functions incorporated in a product design can be integrated to form one or more systems. It is possible that a separate system will be defined corresponding to each of the professional disciplines within the Product Development Department. This means that a single design may incorporate separate systems to be worked out by the mechanical engineers, electrical engineers, pneumatic specialists, hydraulic specialists, software specialists and others. The Product Development Department is primarily interested in ensuring that the separate systems are appropriately integrated.

The market demand for more specific, customized products has increased in recent years. This means that today's Product Development Department needs to design many more finished products than before. This means that the emphasis has shifted from designing individual finished products to designing product families in many Product Development Departments. Product families are designed at the component level as well as at the finished product level. The specific demands of customers can be met more easily when the finished products can be assembled easily from such product families at the component level.

As seen in the preceding chapters, a generic bill-of-material is extremely useful for defining product families. A generic bill-of-material can be used as the basis for recording the available product knowledge by making use of continuous parameters to describe a product in detail. Nevertheless, it is not always feasible or practical in terms of the effort required to structure all of the product knowledge in a manner suitable for recording in the generic bill-of-material. In other words, it is not always possible to determine in advance the details concerning the types of products to be delivered. A generic bill-of-material may provide a certain degree of support in these types of situations. Two types of situations can be identified in particular.

The first type of situation involves product families for which the structures of the finished products differ, but nevertheless finished products that make use of the same generic components. This means that there will be significant differences at the top levels of the bills-of-materials. An example of this could be a product family of conveyor systems. Each conveyor system to be delivered will include a number of conveyor belt subassemblies, all of which may be different. The layout of any particular system will likely be different from any of the other systems. A generic conveyor belt component could very well be defined as a generic bill-of-

material, however a total conveyor system could not be defined as a generic bill-of-material without spending a great deal of time and effort to define all of the possible configurations of conveyor systems.

A second type of situation involves product families in which the finished products are comprised of one or more components that cannot be specified fully using a generic bill-of-material. An example of this could be a product family of bottling machines to fill empty bottles where all of the variants of this product family can be described using a generic bill-of-material. Furthermore, we assume that one component, the filling nipple that attaches the machine to the bottle, cannot be specified fully using the generic bill-of-material. This is because the empty bottles may come in such a wide range of sizes and shapes so that it is virtually impossible to formulate design rules to accommodate every possible type of bottle. This means that this one component cannot be specified fully in the generic bill-of-material.

The generic bill-of-material could, nevertheless, include a basic specification for the nipple. This means that a number of characteristics of the required nipple component can be specified, in particular, those characteristics dealing with the size of the chosen variant of the bottling machine. These characteristics of the component can be described based upon the external parameters of the generic component specification. These external parameters specify how the generic component interfaces with its environment.

Theoretically, a bill-of-material for a specific customer order can be generated, therefore, by four possible processes:

- 1 a configuration process based upon customer order independent product knowledge that is recorded in a generic bill-of-material;
- 2 an assembly process whereby components are built from modules that can be configured using a generic bill-of-material system. Generic bills-of-materials are available in this case for the individual generic modules, but not for the total assembly;

- 3 a detailed specification process based upon a generic bill-of-material for a component that is not yet fully configured;
- 4 a design process for a component, without the use of a generic bill-of-material.

The aforementioned processes are presented in decreasing order based upon the degree to which an automated system can be employed effectively to support the respective processes. The possibilities for automating these processes are discussed in more detail in the next chapter.

Based upon this, there are four parts of a bill-of-material for a specific customer order that can be distinguished (refer to Figure 7.1).

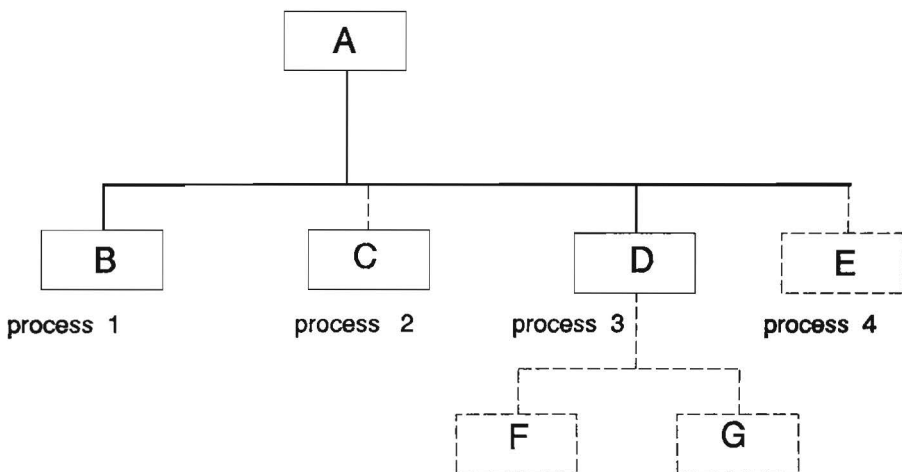


Figure 7.1: Customer order linked bill-of-materials divided by source

A number of benefits can be derived through the use of a generic bill-of-material by the Product Development Department:

- 1 the lead time can be reduced for designing products for specific customer orders as well as for anonymous orders within the Product Development Department;

- 2 standardization can be stimulated through reuse of generic components (i.e., a generic component can be treated as a certain class of components);
- 3 product designs can be improved by keeping track of experience with existing components and recording this in the generic bill-of-material.

### **Manufacturing point-of-view**

The Manufacturing Department is responsible for producing high-quality products and delivering these on time at a minimal cost. Manufacturing will normally split the finished product into a number of separate parts to be produced. It must be possible to record the status of each of these parts. For this reason, each of these parts must appear as a subassembly in the bill-of-material. It is not necessary to keep track of all of the intermediate products in the manufacturing process. Nevertheless, the progress is always recorded at the intermediate stock points in the manufacturing process. This means that this level must always be included as the minimum level in the bill-of-material. In other words, the basic logistic aspects must be clear from the bill-of-material as a minimum requirement.

These levels must also be included in a generic bill-of-material whenever it is used to generate a bill-of-material for a specific customer order.

The problem of how to manufacture a product is not discussed here since the bill-of-material only documents what should be manufactured and not how this is to be done.

The Manufacturing Department will normally wish to produce a number of identical components as a batch, whenever possible. This is controlled automatically by an MRP system when products are made to stock. This is not controlled automatically in the case of customer order driven manufacturing, however. A customer order is normally split into a number of work orders. Each work order generally corresponds to a line item in the bill-of-material. When a given component appears as several line items on the bill-of-material, a decision needs to be made whether to combine these line items and group them into a single work order. This decision can be recorded in the bill-of-material by choosing a different view of the product structure.

The example of the product family of conveyor systems described in the section on the Product Development Point-of-View can be used again to illustrate this. Each conveyor system is designed for a specific customer. As previously indicated, each conveyor system is comprised of one or more conveyor belts. Each conveyor belt is constructed with two or more supports. All of the conveyor belts included within a given conveyor system normally have the same height and width. This means that all of the supports used in a given conveyor system will normally be identical and would be manufactured, preferably, as a single batch. The problem is that these supports will normally appear as separate line items under each of the conveyor belt assemblies on the bill-of-material. If all of the support variants are grouped together via a generic product, then the identical and similar supports belonging to one or more customer orders can be grouped into a single batch.

### **Sales point-of-view**

The Sales Department is interested in describing the range of products it can offer to its customers. Components may be sold as well as finished products. The product offerings will generally be described in a product catalog.

In its simplest form, the product catalog will list the product article numbers with the respective product descriptions. The product listings are organized by category in order to make the catalog easier to use. The product categories are often based upon the product applications or ways in which the products are used.

Other ways of identifying the products can also be used in the catalog. Four different methods of specifying products were described in the previous chapter. Two of these methods make use of parameters, however, the parameters that are useful for describing products for sales purposes are often different from the parameters used for the technical product descriptions needed for manufacturing and/or product development. The parameters needed for sales purposes are often related more to the product application or market sector.

The most important documents for the Sales Department are the orders and quotations. When a product catalog is used, the product article numbers on appearing on these documents should be the numbers used in the catalog. When no catalog is used, much more documentation will need to be included with the

customer order to adequately describe the product to be delivered. Examples of this supplementary documentation could be drawings of the layout and essential components and technical specifications.

The product structure is only important information for the Sales Department to the extent that the product components are used to identify and specify the product. We refer to this as an indirect form of identification. The product components used by the Sales Departments may be different from the components used in manufacturing or in designing and developing the product.

A generic bill-of-material can be especially useful for selling complex products that have many variants. The generic bill-of-material can be used to determine the cost price and the delivery lead time as well as the product specification. This is discussed in more detail in the next chapter.

### **Combined points-of-view**

Ideally, a company would be able to find a single point-of-view for structuring all of its products. In this way, everyone in the company would be using the same "language" to identify and document all of the product information. It is more pragmatic to accommodate multiple points-of-view, however. The major disadvantage of this approach is that the task of maintaining the product information increases exponentially as the number of points-of-view increases. This is because the consistency of the information across all of the points-of-view also must be maintained.

Maintaining the consistency of information across multiple points-of-view generally requires close cooperation between the different departments and agreement regarding the minimum number of products on a bill-of-material list. When a company manufactures components and also assembles the finished products, at least one level of products will need to be defined between the primary products and the finished products. If intermediate stock is held at the point between the manufacturing of components and the assembly of finished products, then this will be the most likely level in the bill-of-material for defining intermediate products. That means that this intermediate level should appear in the various bills-of-materials in addition to the finished product level and the primary product level.

---

Furthermore, all of the different bills-of-materials needed to provide the different points-of-view for a given finished product will need to have at least these three levels.

As a result, any product levels that may be needed in addition to the aforementioned intermediate stock level will need to satisfy a number of conditions.

- 1 any extra subassembly within the assembly stage of the process must include a component at the intermediate level on each of its paths to the primary product level;
- 2 any extra subassembly within the component manufacturing stage of the process must have a single primary product on each of its exploded paths;
- 3 any extra subassembly within the component manufacturing stage of the process must have a single subassembly at the intermediate level on each of its imploded paths.

Several examples are presented in Figure 7.2 that do not satisfy the conditions listed above.

An additional condition must be imposed for generic bills-of-materials. The same specialties must be defined for a generic product as for the various bills-of-materials for the different points-of-view for the different disciplines within a company. Only then can they refer to the same variants when using the generic bill-of-material.

### **Architecture**

The question arises of where a generic bill-of-material system belongs in relation to a company's information system architecture. Figure 7.3 shows the information flow associated with describing products as they pass through the Sales Department, Product Development Department, Production Planning Department and Manufacturing.

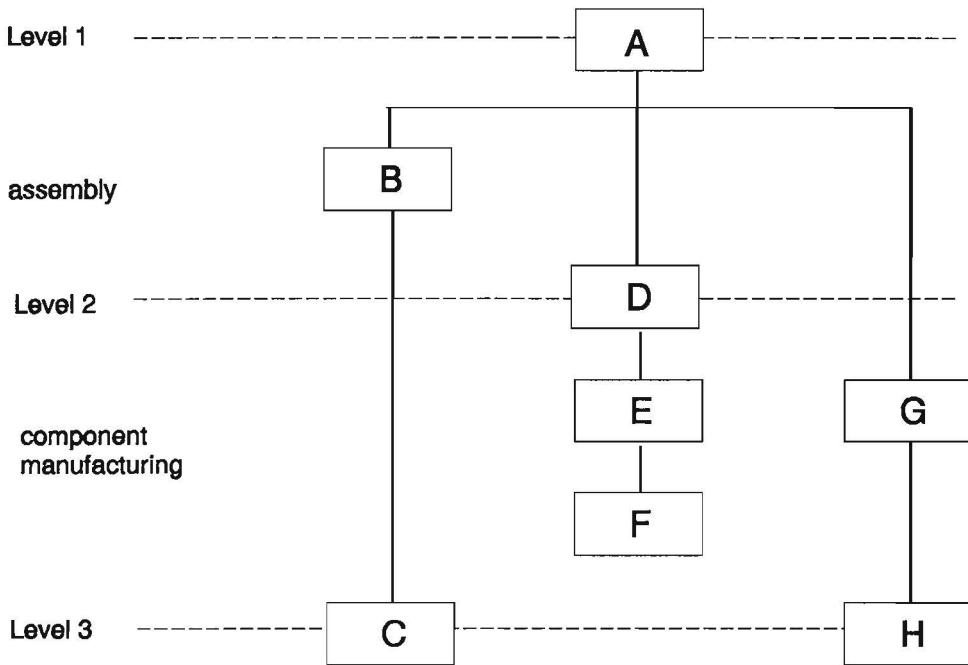


Figure 7.2: Bill-of-materials which does not satisfy the 3-level requirement

These departments often use their own product modelling systems and, sometimes, even their own bill-of-material systems. The information will need to be transformed as required as it passes from one department to the next in this case. The possibility of introducing errors increases whenever information is transformed in this way. This means that the information may not be consistent from one system to the next. For this reason, it is important not to allow the proliferation of multiple bill-of-material systems within a company. The same situation applies to generic bill-of-material systems.



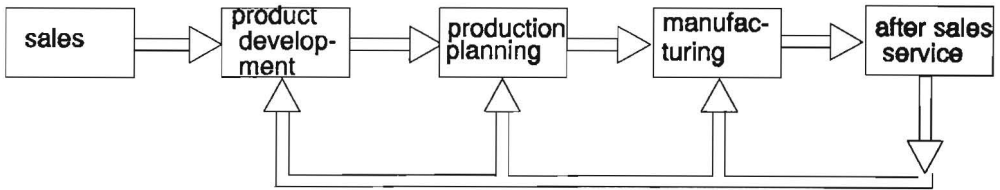


Figure 7.3: Information flow for product description

Ideally, only one generic bill-of-material system will be implemented within any single company. The ideal system would be used at the company level and be able to generate information for all of the departments within the company. A decision must then be made regarding which point-of-view should be used to define the product structures within this type of generic bill-of-material system. This choice of point-of-view depends upon several factors.

The first factor is the frequency of usage of the information recorded in the generic bill-of-material. This factor can be split into the frequency of usage in connection with keeping the information up-to-date and the frequency of usage in connection with information retrieval. The frequency of usage for data maintenance is typically a small fraction of the frequency of usage for information retrieval purposes. The Manufacturing Department will normally be the major user. In this case, there will be a preference for using the manufacturing point-of-view for structuring the generic bill-of-material.

The second factor is the stability of the recorded information. The question to be asked here is how long the same information can be used before it needs to be updated. The first factor normally dictates that the manufacturing point-of-view should be used. In this case, however, the Manufacturing Department could revise the routing information related to the product structures relatively frequently so that the generic bill-of-material would also need to be updated frequently. In other words, a significant data maintenance effort would be required. In order to reduce the need for excessive data maintenance in this case, the generic bill-of-material could be defined in such a way that changes in the routing would not affect the product structure.

If a single system for the whole company is implemented, then it is normally easier to interface this system with the various departmental systems. Examples of departmental systems might be CAD and CAM systems. These types of systems should be set up to support generic products. This means that they should be able to generate order-dependent information based upon input data about customer orders that is structured according to a generic bill-of-material. The quantity of information to be exchanged between the generic bill-of-material system and the departmental systems should be kept to a minimum. This implies that the identification and specification of a configured product variant must provide sufficient input information to the departmental systems to enable them to generate the information necessary for their specific functions. This is summarized in Figure 7.4.

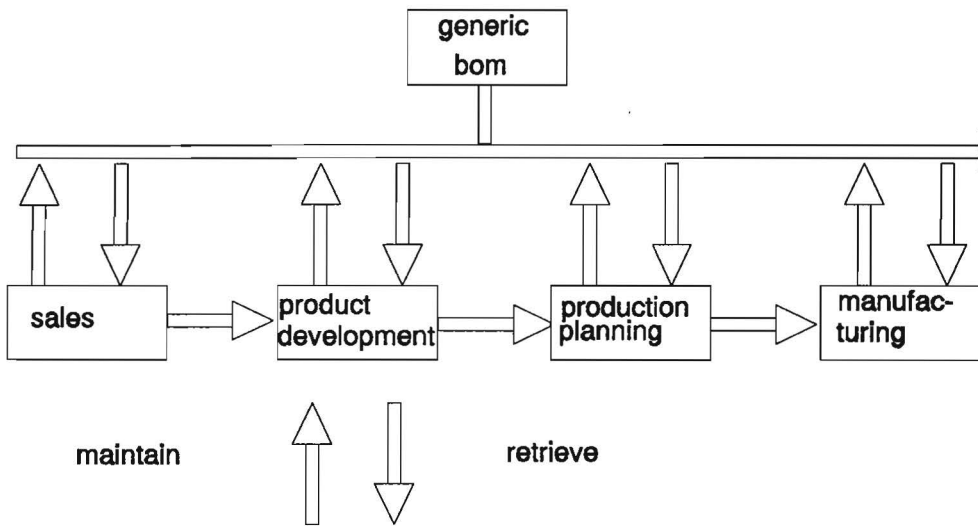


Figure 7.4: Architecture with a single generic bill-of-materials within a company

Another variant for this architecture is implementing a bill-of-material system for one or more departments without a coordinating generic bill-of-material system. These departmental bill-of-material systems may have a generic or specific nature. The problem with this variant is interfacing the different departmental systems. Maintaining the integrity of such interfaces is generally a major effort. Some

agreement is needed concerning the common levels to be used in each bill-of-material in order to reduce this maintenance workload.

Another possibility is to record the agreements concerning the common levels in the bill-of-material in a single common generic bill-of-material system. This system can then be used to coordinate the departmental systems.

We can therefore distinguish the following types of architectures:

- 1 an architecture consisting of only departmental systems;
- 2 an architecture consisting of a single generic bill-of-material system;
- 3 an architecture consisting of departmental systems coordinated by a common generic bill-of-material system.

### **Summary**

In this chapter we described how a generic bill-of-material should be constructed and where we can make use of a generic bill-of-material. The following aspects were addressed:

- which variants should be covered by a generic product;
- which parameters should be used to describe a generic primary product;
- which parameters should be associated with a generic subassembly;
- which generic product should be associated with a condition or an algorithm;
- which point-of-view should be used to define a bill-of-material;
- which levels should be included in a bill-of-material;
- where a generic bill-of-material system should be positioned in relation to the company's total information systems architecture.

Several different architectures were described in the previous subsection. Additional research is needed to determine which architecture is best suited to which environments based upon certain criteria.

### Introduction

An initial application of the generic bill-of-material is described in this chapter. The configuration of a variant here refers to specifying a feasible variant that meets customer requirements. A system used to support this process is called a configurator. A configurator can be used in a number of different ways. For example, a configurator can be used to support the sales process and can also be used by a design engineer to help select and specify previously designed components to be incorporated in the new design. The generic bills-of-material, of course, will be different in each case since they need to support different points-of-view.

### Configurator and bill-of-material generator

A **configurator** is a system that supports the identification of a variant. This type of system provides support that is based upon product documentation. This product information is documented in the form of a generic bill-of-material. The final product of the configuration process is the identification of a single variant or a set of variants. In the latter case, the configuration cannot be considered to be complete since certain components will still need to be specified in detail. It is assumed that the configurator is not able to provide a sufficiently detailed identification in this case.

A **bill-of-material generator** produces a bill-of-material based upon the identification provided by a configurator. When the identification is incomplete, the generated bill-of-material will also be incomplete since the bill-of-material generator uses the information incorporated in the generic bill-of-material. This situation is illustrated in Figure 8.1.

A variant identification can also be used for other purposes. For example, this can be used as the basis for generating other product documentation such as drawings, assembly instructions, cost price calculations, programs for numerical control and service manuals. The systems used to generate other types of documentation are driven by the variant identification that is produced by the configurator.

## Identification

As indicated above, the final product of the configuration process is the identification of a variant. The identification of a variant produced by a generic bill-of-material may fall into one of the following three categories (refer also to Chapter 5):

- 1 direct identification without parameters
- 2 direct identification with parameters
- 3 indirect identification with parameters.

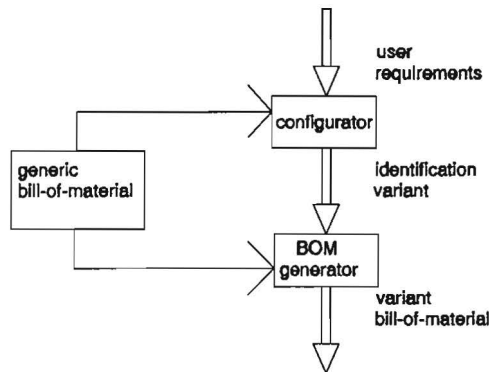


Figure 8.1: Configuration and bill-of-material generation process

### Direct identification without parameters

A direct identification without parameters can be achieved through the use of a product code assigned to the generic product. For example, the seat frame in our chair example was identified by product code 3265.

### Direct identification with parameters

A direct identification with parameters is based upon the identification of the generic product specialty associated with the configured variant at the lowest level. The generic product may be a finished product, an intermediate product or a primary product. As seen previously, the identification of a specialty within a generic bill-of-material can be accomplished by specifying the product code of the

---

GP and the sequence number of the specialty. An example of this is the identification of a stand (swivelling, without wheels) in terms of the GP product code 7291 associated with the stand and specialty sequence number 1.

#### Indirect identification

An indirect identification with parameters is the most complex form of identification since the decision path must be specified in addition to the generic product and specialty at the lowest level.

A variant of a chair is illustrated in the form of a product structure in Figure 8.2. This variant can be identified indirectly using parameters. Four choices need to be made in this case to arrive at a specific variant. These choices are:

- 1 choosing red for the chair GP
- 2 choosing without wheels for the underframe GP
- 3 choosing swivelling for the stand GP
- 4 choosing without armrest for the armrest GP.

These choices are indicated in parentheses in Figure 8.2 and can be found on two separate paths in the product structure. These paths are called the decision paths. The nodes associated with the first path are the chair, underframe and stand generic products. The second path can be identified by the chair and armrest GP nodes. The decision process dictates that the GP's found on the decision paths must be replaced by one of their specialties at the lowest level. As an example, the underframe GP is replaced by the lowest level specialty that specifies the variants without wheels. This lowest level specialty actually includes two possible variants, the swivelling and the non-swivelling variants. The wheel component is included in Figure 8.2 to complete the picture, however, the wheel is not actually a component of this chair variant. For this reason, the wheel component is shaded.

The aforementioned decision paths provide the basis for identification and can be represented as a decision tree. This type of decision tree is illustrated in Figure 8.3a for the example of a chair. A line number is assigned to each line of this type of identification so that each line can be uniquely identified. The tree structure, thus, can be represented by including the line number of the parent. The parent line number identification indicates which parent is associated with each lowest level

specialty component. The position number is the sequence number in the bill-of-material relationship.

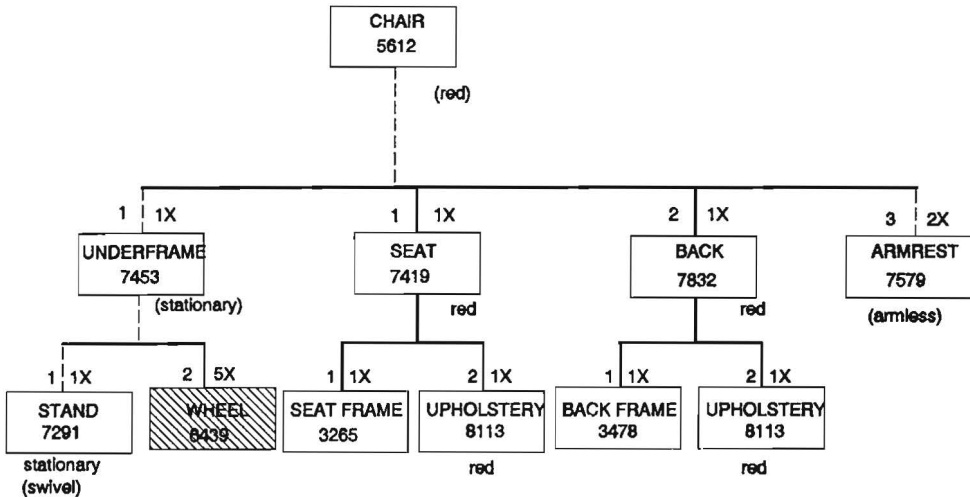


Figure 8.2: A chair variant (red, stationary, swivel, armless)

line number	parent line number	position number	generic product number	lowest level specialty number
0			5612G	1
1	0	1	7453G	2
2	1	1	7291G	1
3	0	4	7579G	3

Figure 8.3a: Identification tree for a chair variant

A variant will normally be used many times during its lifetime. The identification of a variant is required, for example, to be able to prepare a quotation, customer order or work order, or for controlling and allocating stock. For this reason, it is useful to assign a key to each identification tree. Such a key can be comprised of a GP product code, a specialty number and a variant number. The GP product number and the specialty number are needed to specify an initial specialty for the configuration process. The variant number must be unique for a given specialty. The variant number is normally a temporarily assigned number that can be

generated automatically. The variant number and the associated identification tree may be deleted automatically after a specific period of time in certain situations. This may occur, for example, when none of the planning data and none of the material transactions processed by the total information system refer to a given variant.

generic product number	lowest level specialty number	description	parameter	parameter value
5612G	1	chair	COLOR	RED
7453G	2	underframe	WHEELS	NO
7291G	1	stand	WHEELS	NO
			SWIVEL	YES
7579G	3	armrest	ARMREST	WITHOUT

Figure 8.3b: Lowest level specialties in an identification tree

The aforementioned type of identification can be described using the data structure diagrammed in Figure 8.4. This data structure diagram shows two entity types: a variant identification entity type and an identification relationship entity type. The variant identification entity includes several attributes: the GP product code, the specialty number and the variant number. The identification relationship entity type has the attributes listed in Figure 8.3a.

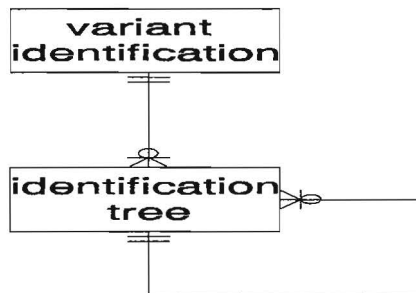


Figure 8.4: Data structure for indirect variant identification

The composition of the key for each different types of identification is summarized in Figure 8.5.



The data structure can be improved to some extent to eliminate data redundancy. Redundant data is apparent in a situation where variants are identified indirectly. A variant in this case will usually have components that are also identified indirectly and, thus, also have an identification tree to define the relationships. This tree will be identical to a sub-tree of the parent's identification tree. This means that certain information will be recorded twice. This occurs because information about the components as well as about the finished product is needed to manufacture the finished product.

identification	type	generic product number	lowest level specialty number	variant number
direct	without parameters	key		
direct	with parameters	key	key	
indirect	with parameters	key	key	key

Figure 8.5: Composition of the key for each type of identification

This can be illustrated using the example presented in Figure 8.2. Assume that a underframe variant first needs to be manufactured in order to assemble the chair variant in this example. A work order must be issued to have the required swivel underframe without wheels built. An identification code for this underframe variant must be included on this work order. The require identification tree is equivalent to line numbers 1 and 2 of the identification tree for the complete chair variant.

It is also possible that a component variant like this will be incorporated in other finished product variants. This situation is similar to using a normal bill-of-material (see Figure 8.6). The data structure typically used for a bill-of-material is well-suited for use in recording the necessary information in this case. There is one major difference, however. A variant with a direct identification with parameters is found at the end of each decision path rather than a variant with a direct identification without parameters. This is because we are dealing with a configuration process here in which a variant with an indirect identification is to

be chosen. The results of all of the decisions are associated and stored with the entity types shown in the data structure presented in Figure 8.6. This variant, therefore, does not occur in the form of the variant identification entity since this entity type only covers the variants that are identified indirectly such as the chair and the underframe. The stand and the armrest are excluded in this case. This means that the implosion relationship is a 0,1 to n relationship instead of a 1 to n relationship as in the case of a bill-of-material.

The chair example presented in Figure 8.2 incorporates two decision paths, namely: (chair - underframe - stand) and (chair - armrest). The stand and the armrest are found at the end of these decision paths. The associated variants are identified directly with parameters.

This situation is illustrated in Figure 8.6. The entity types for the variants and choice relationship in this data structure correspond to the product information and bill-of-material relationship in a bill-of-material (Figure 4.7).

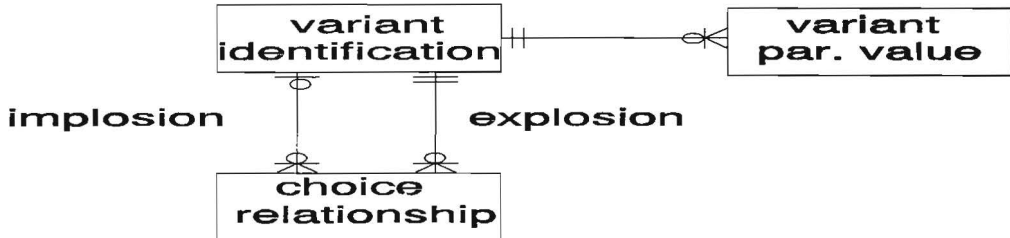


Figure 8.6: Improved data structure for the variant identification including continuous parameter values

#### Identification with continuous parameter values

A facility for including a continuous range of parameter values in a generic bill-of-material was presented in Chapter 6. These parameter values are selected as part of the configuration process. The chosen continuous parameter values become a part of the variant identification and, thus, need to be included in the documentation associated with identifying the variant. Since these parameter values are used in connection with the direct identification with parameters as well as with

the indirect identification, the direct identification with parameters also needs to be included in the documentation associated with identifying the variant. The attributes associated with each type entity are summarized below.

Variant identification	Choice relationship	Variant par. value
GP product code	position number	GP product code
Specialty number	PARENT GP product code	Spec. number
Variant number	PARENT spec. number	Var. number
	PARENT var. number	parameter
	COMP. GP product code	par. subrange
	COMP. spec. number	par. value
	COMP. var. number	

The consequence of including a chosen continuous value in the documentation is that the paths need that lead to where this decision is made need to be identified and documented. This normally means that more paths will need to be documented in this case as opposed to when there are no continuous parameter values.

It is not necessary to document chosen parameter values when the variants have only non-continuous (discrete) parameter values. This is because the parameter values are known from the fact that specific lowest level specialties are chosen.

### **Configurator based upon a generic bill-of-material**

Decisions need to be made when a product variant is configured. The decisions are documented in a structured manner when a variant is configured based upon the concept of a generic bill-of-material. Two types of decisions can be identified:

- 1 choosing from the available specialties or, in other words, choosing a subset of variants
- 2 choosing continuous parameter values.

These decisions are made in a "depth-first" sequence. This means that the first decision is taken at the GP level for the GP associated with the variant to be configured. Then the generic components in the left-most branch of the generic bill-of-material are considered. This process continues from left to right through all of the branches in which decisions need to be made and until there are no further decisions to be made.

---

An alternate approach is based upon the "breadth-first" sequence. With this approach all of the decisions are made at the same level in the identification tree. This means that the component identifications are carried out simultaneously. Further research would be required to determine which of these two approaches is actually better. In the configuration process described here, the "depth-first" sequence will be followed.

Four basic processes can be identified in connection with a configuration process based upon a generic bill-of-material:

- 1 choosing the top level specialty at the beginning of the configuration process
- 2 traversing a generic product structure
- 3 traversing a decision tree
- 4 inheriting.

### **Starting with the top level specialty**

Three situations are possible at the beginning of the configuration process that refer to the number of top level specialties involved in the initial GP:

- 1 multiple top level specialties, e.g., when a underframe needs to be configured such as in our chair example
- 2 a single top level specialty
- 3 no top level specialty; the initial GP may then consist of one or more variants. In the case of multiple variants, the choice of variants will then take place at a lower level in the generic bill-of-material.

A top level specialty, as with any other specialty, can be characterized in terms of discrete parameter values and continuous ranges of parameter values. In the case of a continuous range of parameter values, an additional parameter value will need to be chosen within the specified range.

If there is only a single top level specialty, then this may only be a universal top level specialty. This is a specialty that incorporates all of the variants belonging to a given generic product.

## Product structure

The second basic process is the process of traversing the generic product structure. As explained in Chapter 4, a product structure is essentially a set of parts lists. A parts list has been defined as the set of all of the bill-of-material relationships belonging to a given parent. Three subprocesses are required to traverse a parts list:

- 1 start traversing a new parts list
- 2 go to the next bill-of-material relationship in the parts list
- 3 finish traversing a given parts list.

A recursive function can be defined based upon the above-mentioned subprocesses and used to traverse a product structure. An example of this function (BOM) can be formulated as follows:

BOM(parent, quantity):

until all of the parent bill-of-material relationships have been processed

read the next parent bill-of-material relationship

BOM(component of the bill-of-material relationship, quantity \* quantity-per  
for the bill-of-material relationship)

END BOM

## Decision path

The third basic process is the process of traversing a decision tree. Generally speaking, traversing a decision tree is performed using the same subprocesses as in traversing a product structure. However, there is one obvious difference between the two basic processes. There is a difference between the results of the processes. The result of traversing the product structure is a bill-of-material. The result of traversing a decision tree, on the other hand, is a lowest level specialty. The lowest level specialty is found at the end of a decision path. Each path starts at the top level specialty.

This affects the details of how the subprocesses are defined. The basic functions are the same as those used to traverse the product structure. The second subprocess must be modified, however. A choice relationship must be chosen first in this process, before performing the next CHOICE process.

---

If a specialty with continuous parameters is selected as the result of this decision process, then a value must be assigned to each of these continuous parameters.

### **Continuous parameters**

Values may be assigned to continuous parameters that are not associated with top level specialties, in three ways:

- 1 by inheriting a value from a super-specialty
- 2 via an algorithm
- 3 by direct assignment of a value by the user.

When a sub-specialty has the same continuous parameter as its super-specialty, the associated value is transferred automatically. This is only possible when this value falls within the subrange of the sub-specialty's parameter. If this condition is not met, then this sub-specialty is not a valid choice. In this way, the degrees of freedom of the CHOICE process are limited.

The second way in which a continuous value may be determined is through the use of an algorithm. Such an algorithm is associated with a super-specialty and calculates new parameter values based upon the parameter values of the super-specialty. The calculated values are then passed on to the sub-specialty. The process of transferring such calculated values is similar to the inheritance process described above.

The first and second ways of assigning values for a given continuous parameter may produce different results. In this case, the use of an algorithm will have priority over the first method.

This can be illustrated using the example of the box (Figure 6.12) in which the generic side component has a top level specialty and a lowest level specialty. One of the continuous parameters is the width. This width parameter exists for the top level specialty as well as for the lowest level specialty. The width of the top level specialty is inherited from the box. The width of the lowest level specialty is calculated using an algorithm. The calculated width has priority over the inherited width, the width of the top level specialty.

After the continuous parameter values are determined through inheritance or through calculations, it is still possible that some of the continuous parameters may not yet have a value. These values must then be specified by the user. The specified parameter values must, of course, fall within the sub-specialty's subranges for the respective parameters.

The order in which the user of the sub-specialties related to a given super-specialty needs to specify such parameter values is dictated by the order of the choice relationships. The order of specifying these values may also be determined by an attribute of the choice relationship. This type of attribute is similar to the position number in the bill-of-material relationship. A choice percentage could also be used for this attribute. This choice percentage is based upon a ratio with respect to super-specialty X where the numerator is the number of times the choice relationship associated with this choice percentage has been chosen and the denominator is the total number of decisions made. The most popular choices are considered first in this way.

### **Inheriting**

The fourth basic process, the process of inheriting, is defined as the process of transferring chosen parameter values from the parent to its component. These parameter values are values associated with of the chosen lowest level specialty parent. They need to be passed on to a top level specialty of the component. A connection must be established between a lowest level specialty and a top level specialty before a value can be inherited. The question must be answered of which top level specialty to choose.

The top level specialty must satisfy three conditions (refer also to Chapter 5):

- 1 the parameters of the top level specialty must be a subset of the parameters of the lowest level specialty
- 2 for each discrete parameter associated with the top level specialty, the parameter values for the lowest level specialty must be a subset of the parameter values for the top level specialty
- 3 for each continuous parameter associated with the top level specialty, the subranges of the parameters of the lowest level specialty must be a subset of the subranges of the parameters of the top level specialty.

---

The parameter values assigned to continuous parameters are transferred from the lowest level specialty to the top level specialty in the case of inheriting.

A top level specialty is considered to be universal when only one top level specialty exists for a given component and this top level specialty does not have any parameters. This means that a universal top level specialty covers all of the variants of a given component. In addition, the selected lowest level specialty of the parent may include each of the variants of the component. In other words, the component's environment has not imposed any restrictions with respect to choosing a component variant in this case.

To illustrate this, our chair example can be used. The complete generic bill-of-material is presented in Figure 6.2. The armrest has a universal top level specialty that incorporates all of the variants of the armrest. Any of the lowest level specialties of the chair parent may be combined with any of the variants of the armrest.

An integrated model of the aforementioned basic processes is described in the Appendix. The configurator used here will be referred to as the simple configurator.

### **Alternative configurators based upon a generic bill-of-material**

There is one major disadvantage when using the simple configurator. The disadvantage is that all of the decisions need to be made from the beginning for each configuration. This is especially bothersome when only minor changes need to be made to existing configurations with a large number of parameters. An example of this is when a certain variant is used as the basis for configuring a new variant that is similar. In practice, only a limited number of basic variants are typically offered to the customer. The customer then chooses a specific basic variant and decides which changes need to be made to satisfy his specific requirements. Only a limited number of additional choices are made in this way.

A flexible configurator is designed to be used in this way. This type of configurator can be used to start the configuration process based upon an initial configuration. An initial configuration is specified by providing a variant identification that may or may not be complete. This means that the initial configuration may describe



multiple variants. If an initial configuration is not specified, then we return to the situation in which the configuration process must cover all of the decisions since each part of the variant identification will need to be determined. All of the parameters will need to be specified.

Three levels can be identified for using a flexible configurator. There is a separate process at each level, as follows:

- 1 specify a complete identification tree starting from the product structure of the initial GP
- 2 select a generic component K in the complete identification tree
- 3 specify a variant of the generic component K.

These three levels are illustrated in Figure 8.7.

A complete identification tree for the generic product used to configure the desired variant is specified at the first level. A **complete identification tree** consists of all of the feasible identification paths in the product structure of a specific generic product. Subsequently, the initial configuration is specified with respect to this complete identification tree. An example of an identification tree is presented in Figure 8.3a. All of the nodes of the identification tree specify a lowest level specialty in this example. The initial configuration is fully specified in this case, thus, describing only a single variant. If the initial configuration had not been fully specified, then this initial configuration would have described multiple variants. The identification tree would have had one or more empty nodes in this case.

One node in the tree may be chosen at the second level. In other words, a generic component can be selected in the identification tree. If we assume that this is generic component K (see Figure 8.7), then the next decision is to determine which lowest level specialty is required for generic component K (e.g., low level specialty 5 in Figure 8.7). One of two alternative search procedures may be followed to carry out this process: searching based upon a single level explosion or searching path by path. The path in this case starts at the initial GP, leads to the generic component K and may continue to the nodes of the left-most branch(es) of generic component K. This is not possible in Figure 8.7 since K is found at the lowest level in the identification tree.

If a generic component K is selected at the second level, then this can be specified at the third level. This means that a lowest level specialty will be selected for this generic component K. This is accomplished based upon the full decision tree. As explained in Chapter 6, a full decision tree for a GP is created by combining several decision trees. The decision trees to be combined are those that belong to all of the GP's included on the path beginning at the initial GP and ending at the generic component K to be specified. The inheritance mechanism is used to combine the various decision trees. The full decision tree for the underframe for our chair example is illustrated in Figure 6.3. It is important to note that a full decision tree is always based upon the choice of a single generic component. A variety of full decision trees, all different, are typically identified and used during the configuration process.

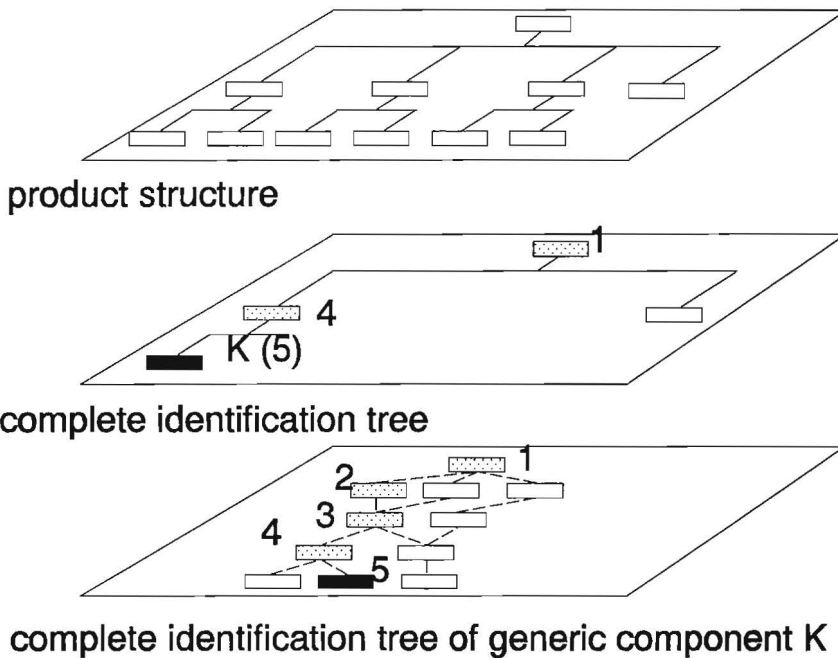


Figure 8.7: The three levels of the flexible configurator

The full decision tree is used at level three. A decision path is identified in this decision tree, ultimately leading to the chosen lowest level specialty 5 of generic component K in our example. A different lowest level specialty can be selected by changing the decision path. This may have implications for the chosen lowest level

specialties for the generic products found at higher levels in the product structure. When decisions are changed in this way, it is important to realize that such changes need to take all of the previously made decisions and choices into account. If a different lowest level specialty is chosen for a given GP, then this may have implications for the generic components of this GP that lie on other paths.

As mentioned above, the full decision tree is used at level three. An expert system could be used in the place of our procedure that is based upon this decision tree, however. As explained in Chapter 6, the required product identification could be derived from the information documented within the generic bill-of-material system. Alternatively, decision tables could be used (see Chapter 6).

The aforementioned approaches for the processes to be carried out at level three could also be applied to the total problem. This implies that the full decision tree for all of the relevant GP's would be combined into a single, large decision tree. The disadvantage of this would be the increased complexity; the advantages of the straightforward decomposition of a generic bill-of-material would be lost.

### **The configurator and order acceptance**

The traditional application for the configurator is to support the customer order acceptance process. Three types of information play an important role in the order acceptance process:

- 1 the product to be delivered
- 2 the delivery due date;
- 3 the price to be charged.

The configurator provides a great deal of support with respect to the first type of information. This is certainly the case when the products are defined and described in terms that the customer understands. The delivery due date will typically be dependent upon the customer requirements. The customer requirements may also be influenced by the quoted delivery lead times for various alternatives. Similarly, the price may also influence the choice of variant.

If use of the configurator is included as part of the Material Requirements Planning process, the choice of variants could then take the availability of materials and the desired delivery date into account. The delivery lead time could then be more flexible. This will be discussed in more detail in the next chapter.

If it is necessary to take the cost of a variant into account during the configuration process, then the cost price information will need to be included in the generic bill-of-material system. The GP's and the specialties in the generic bill-of-material are generally sets of products. It is normally difficult to provide a cost price for a set of products, however, it is often possible to determine a minimum and maximum cost price. It should also be possible to provide median and average cost prices when choice percentages are used in connection with the choice relationships. In addition, it is necessary to keep track of the processing costs at each level in the generic bill-of-material.

Appendix The Simple Configurator

BOM(parent, lowest level specialty, quantity):

```

until all of the parent bill-of-material relationships have been processed
  read the next parent bill-of-material relationship
  if the parent includes specialties, then
    top level specialty = MATCH(parent, lowest level specialty,
                                component)
    if a top level specialty is present, then
      lowest level specialty = CHOICE(component, top level
                                      specialty)
      BOM(component, lowest level specialty, quantity x quantity
          per)
    otherwise
      BOM(component, 0, quantity x quantity-per)

```

END BOM

MATCH(parent, lowest level specialty, component):

```

until a top level specialty is identified
  read the next top level specialty
  if no more top level specialties are present then
    match=0
  if the top level specialty is universal, then
    match=top level specialty
  otherwise
    perform inheritance test
    if inheritance is possible, then
      match=top level specialty

```

END MATCH

---

```
CHOICE(generic_product, super):
if super has one or more subs, then
    until a choice has been made or super has no sub
        read next choice relationship of the super
        if no choice relationship found then
            read the first choice relationship of the super
        transfer the continuous parameter values from the super to the sub
        perform optional algorithm and transfer the calculated parameter
        values to the sub
        request input of parameter values for this sub from the user
        if choice is made, then
            lowest level specialty = CHOICE(generic_product,
            sub_speciality)
otherwise
    lowest level specialty=super

choice=lowest level specialty

END CHOICE
```

### Introduction

A second application of the generic bill-of-material is described in this chapter. This application involves using a generic bill-of-material to coordinate materials in an assemble-to-order production environment. This type of use is referred to as a **generic MRP (GMRP)**.

To start with, the term "Assemble-to-order" is defined in more detail in this chapter using a typology of production situations. After the different types of production environments are identified, the basic GMRP concept is introduced and developed. The problems typically encountered by companies when they change from a make-to-stock situation to an assemble-to-order situation are then described. These problems are typically found in four different areas:

- 1 acquiring or manufacturing the components needed for the customer orders
- 2 accepting customer orders
- 3 assembling customer orders
- 4 dealing with uncertainty.

We continue by describing how GMRP can be used to support the decision-making processes in these specific areas. We then explain how GMRP can be used to support a company that has a mixture of make-to-stock and assemble-to-order processes.

### Production environment

As previously seen in Chapter 1, a production environment can be classified in various ways according to different points-of-view. From a logistics point-of-view, a production environment is often classified based upon the location of the customer order decoupling point (CODP). Four types of production environments can be identified in this way [HOEK87][SARI81]. These four categories are listed in Figure 9.1 along with an indication of where the CODP is located in each case.

Choosing the best type of production environment to be used in a given situation will also depend upon the acceptable delivery lead time in the market being served.

	Type	Location of the CODP
1	Make-to-stock	finished product stock
2	Assemble-to-order	component stock
3	Make-to-order	raw materials stock
4	Engineer-to-order	prior to product development

Figure 9.1: Types of production environments

This chapter focuses primarily on the **assemble-to-order** (ATO) production environment. The ATO environment is first compared to the straightforward situation found in a **make-to-stock** (MTS) environment to provide a basis for describing the specific aspects which are different.

### **Assemble-to-order production environment**

The interaction between the market and the production processes are minimal in an MTS environment. The decoupling point is located immediately following the production process. In an ATO environment, the customer order affects the production processes much more since product assembly is driven by the customer orders. This implies that the delivery lead time for customer orders will be greater than in an MTS environment.

One of the reasons for moving from an MTS environment to an ATO environment is to increase the number of possible variants within a product family without incurring the cost of keeping all of the possible variants in stock. Increasing the total number of variants also implies a reduction in the production volumes for each finished product variant and an increase in the uncertainty of product demand.

Offering a larger range of variants at the customer order acceptance point in an ATO environment also implies that the task of product identification is more important than in an MTS environment.

Components need to be manufactured based upon the forecasts of market demand and products need to be assembled based upon the customer orders in an ATO



---

environment. In an MTS environment, however, market forecasts drive both the manufacturing of components and the assembly of products.

Wemmerlöv discovered four core problems which occur when the switch is made from an MTS environment to an ATO environment [WEMM84]:

- 1 acquiring and manufacturing the components needed to assemble the customer orders
- 2 the interaction between production control and the product demand
- 3 assembling the customer orders
- 4 the uncertainty of demand.

These four problems are described in more detail in the following subsections. A number of aspects are common to several of these problems. In addition, we will indicate how a generic bill-of-material could be used to address and find solutions for these problems. This will provide the basis for introducing the concept of using a generic bill-of-material for coordinating materials, or generic MRP (GMRP).

### **1 Acquiring and manufacturing components**

Two characteristic elements of a production control system in an ATO environment are the **Master Production Schedule (MPS)** and the **Final Assembly Schedule (FAS)**. The FAS will be explained later. Among other things, MPS coordinates the production and the sales through the use of a sort of contract agreement. On the one hand, sales is obligated to move an agreed quantity of products. On the other hand, production is obligated to manufacture an agreed quantity. In many cases, these two quantities are not the same. MPS directs the flow of materials and products. This means that MPS drives the acquisition and manufacturing of components in an ATO environment. In this way, sufficient components should be available in future periods to assemble the expected customer orders. In addition, MPS will schedule sufficient capacity to manufacture these components and to assemble the customer orders in an ATO environment.

One of the problems for MPS is the difficulty in forecasting the quantities of finished products that need to be produced in view of the large variety of finished products in an ATO environment. For this reason, forecasting will generally be done at the product family level when there are a large number of product variants.

A **modular bill-of-material** [MATH82][VEEN91] is often used to translate forecasted quantities at the product family level to forecasted quantities of components. This concept has already been described in Chapter 2 and will be reviewed briefly here. Past experience using this approach shows that problems are typically encountered that are similar to the problems discussed in Chapter 2.

A modular bill-of-material approach prescribes that the components in a product family need to be grouped in so-called **planning modules**. Groups are formed based upon the various **features** and **options** of the respective product families. Features and options are similar to parameters and parameter values.

A planning module is viewed in the same way as a product, called a **pseudo product** [ORLI72]. The product family, as such, is also treated as a pseudo product. A pseudo product is, thus, a set of products that is included in a modular bill-of-material. The **planning percentage** is defined as the quantity-per of the bill-of-material relationship between the product family and one of its planning modules. This percentage is the percentage of finished products within the product family for which this planning module is chosen and, thus, its components are chosen.

The components are connected with their respective planning modules via a bill-of-material relationship in the planning bill-of-material. The quantity-per associated with a relationship is determined by the number of times that one component is used in a planning module.

This can be illustrated using our chair example. Assume that this product family consists of only two variants: a red chair and a blue chair. The underframe, seat and back are combined in a subassembly. Subsequently, this subassembly is used with the armrest components to assemble the chair. The CODP is located prior to the subassembly process.

This example involves a feature called "COLOR" and two options called "RED" and "BLUE". This leads to the creation of two planning modules called "RED" and "BLUE". The common stand, wheel, seat frame, back frame and armrest components are included in the "COMMON" planning module with a planning

percentage of 100%. The modular bill-of-material for this example is presented in Figure 9.2.

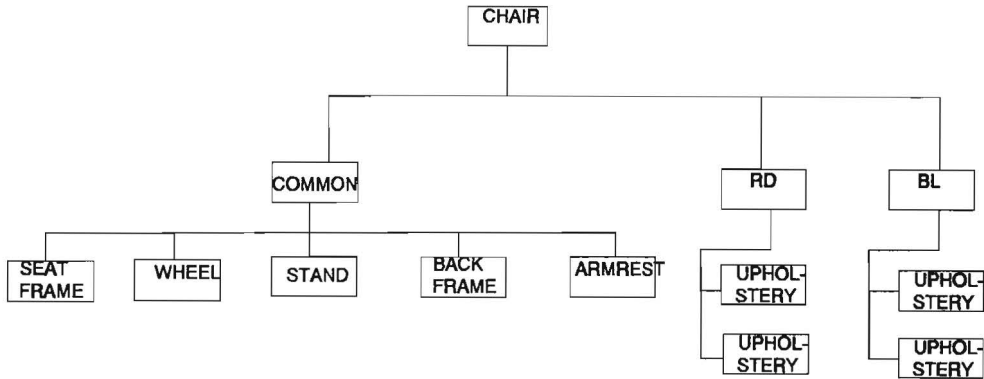


Figure 9.2: Modular bill-of-material for a chair

Two different techniques can be used to forecast the required quantity of components. The first technique is to use the forecast at the product family level in its entirety, and then to use planning percentage estimates as the basis for the forecasts for the planning modules. The second approach is to provide forecasts, directly, for each of the planning modules. When the planning percentages are stable, both of these approaches perform equally well [WIJN87] [GIES93]. When there is little historical data to use as the basis for generating sufficiently reliable planning percentage estimates, the historical data for similar product families could be considered for use in this way.

The major advantages of using a modular approach are a reduction in the number of planning items and a significant reduction in the number of bill-of-material relationships. The administrative effort needed in connection with planning is greatly reduced when the number of planning items is reduced.

In addition to the advantages, there are also several disadvantages in using a modular bill-of-material. One disadvantage is that any information about the product structure and the assembly sequence will disappear. This was first mentioned by Orlicky [ORLI72]. This shortcoming is typically resolved by adding

a so-called **manufacturing bill-of-material** for one or more of the subassemblies. This solution has been recommended by Sari [SARI81].

A second disadvantage is that an engineering change may require changing the structure of the bill-of-material. This happens, for example, in situations where a "COMMON" component is no longer used in all of the finished product variants. This means that such a component must be removed from the "COMMON" planning module and added as a component for a specific option. Another example of when an engineering change can affect the structure of a modular bill-of-material is the addition of a new feature or option.

A possible third disadvantage is when the planning percentages do not remain constant and stable from period to period. This implies that it may be necessary to use multiple planning percentages within a single planning horizon. This shortcoming could be resolved by defining multiple bill-of-material relationships with different percentages and assigning effective from/to dates to each of the bill-of-material relationships.

Finally, it is also possible that an option is in no way consistent with a given subassembly. This means that a combination of various options for different features may be required to arrive at the planning percentage for the components. This normally creates a number of problems [VEEN91].

### **Generic bill-of-material and planning percentages**

A bill-of-material relationship in a planning bill-of-material essentially presents an opportunity for making a choice in specifying a finished product variant. This is represented by the choice relationship in a generic bill-of-material. As seen earlier, the choice relationship is the relationship between a super-specialty and a sub-specialty. Similar to the bill-of-material relationship in the planning bill-of-material, this is a relationship between two sets of products.

A planning percentage may also be associated with a choice relationship, similar to the situation with a planning bill-of-material relationship. The planning percentage indicates the probability of choosing a sub-specialty for a given super-specialty. In other words, the expected number of variants of the sub-specialty to

---

be produced in a given period is equal to the planning percentage times the number of variants of the super-specialty to be produced.

The generic bill-of-material for the example in Figure 9.2 is presented in Figure 9.3. In contrast with the modular bill-of-material approach, the information about the product structure and the assembly sequence is retained when a generic bill-of-material is used.

The second disadvantage described above of using a modular bill-of-material is that an engineering change may require a change in the bill-of-material structure. The implications of this type of change with respect to a generic bill-of-material can be illustrated as follows. Assume that we are using the example of a "basic" chair and that the product family is defined as indicated in Figures 9.2 and 9.3. Next, assume we wish to change this product family to the product family as defined in Figure 9.4. (Note that this is the same diagram as Figure 6.2, but now with planning percentages.)

The engineering change can be summarized as follows:

- 1 There are three colors instead of two colors. This means that an additional GRAY planning module will need to be defined for the modular bill-of-material with gray upholstery components for the seat and the back.
- 2 ARMREST has been added as a feature. This means that since the armrest is not a COMMON component for all variants supported by the modular bill-of-material, this will need to be included as a component in connection with a new planning module called WITH ARMREST.
- 3 Two features have been added with several restrictions. The features are SWIVELLING and WHEELS. As previously seen in Chapter 6, the relevant restrictions are:
  - a. all of the variants with wheels also swivel
  - b. all of the blue variants have wheels.

This has significant implications for the modular bill-of-material. Only two of the COMMON components remain: the seat frame and back frame. The other components need to be assigned to planning modules. Some of the modules to be defined are dependent upon a combination of options. The following modules need to be added:

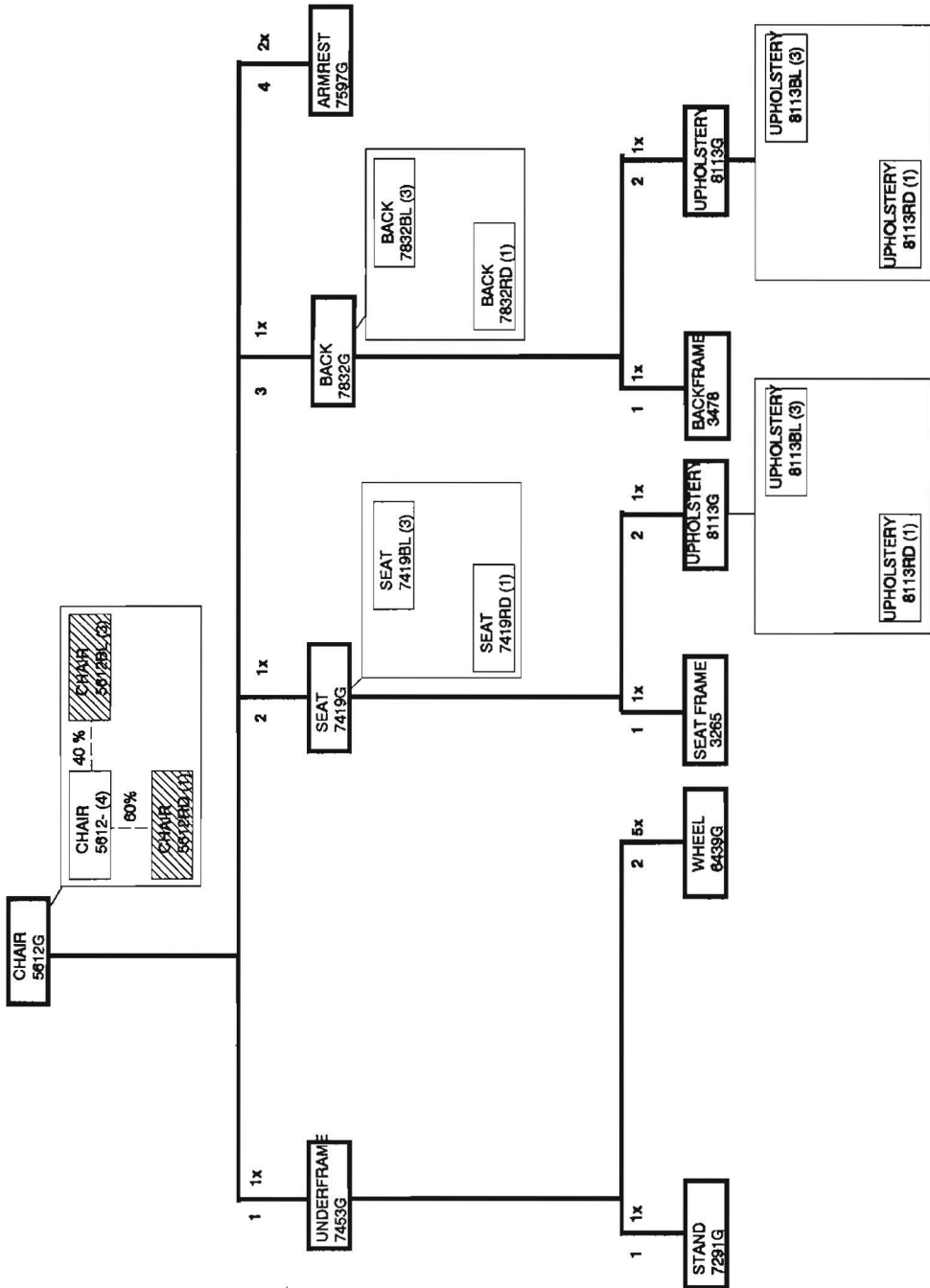


Figure 9.3: Generic bill-of-materials for a "basic" chair

- a. a SWIVELLING WHEELS module with wheels and a swivelling stand component for attaching wheels;
- b. a SWIVELLING GRAY OR RED WITHOUT WHEELS module with a swivelling stand component without wheels;
- c. a NON-SWIVELLING GRAY OR RED WITHOUT WHEELS module with a non-swivelling stand component without wheels.

The modular bill-of-material will be compared to a generic bill-of-material in Chapter 10.

The choice relationships may occur at multiple levels in the generic bill-of-material for a generic product. They may appear in the decision tree of a GP's generic parents as well as in the decision tree of the GP, itself. Planning percentages are included with all of the choice relationships.

The extended example of the chair GP presented in Figure 9.4 includes a large number of choice relationships. The planning percentages are also indicated in Figure 9.4. The expected quantity of stands needed for swivelling chairs with wheels can be calculated automatically using these planning percentages, the forecasted sales of chairs and the quantities documented in the bill-of-material. This expected requirement for stands will be 360 units if the sales forecast for the chairs is 1000 ( $= 0.5 * \underline{1} * 0.8 * \underline{1} * (0.6 + 0.3) * 1000$ ). The underscored numbers here are the quantities per.

It is important to note that multiplying the planning percentages in this way is only valid when the decisions are independent of each other. Wijngaard refers to this data independence as the marketing modularity [WIJNG87].

## 2 Production control and product demand

The following aspects are important in connection with selecting a finished product variant during the process of accepting a customer order in an ATO production environment:

- production feasibility
- price





- 
- delivery lead time (availability check for components, subassemblies and assembly capacity).

When the availability of sufficient components is checked, it is not necessary for all of the components to be present before the assembly of the customer order can be initiated. It is possible that certain components will be required later on in the assembly process. As a result, these components do not need to be available immediately when there is sufficient time to buy or manufacture these parts.

The required components need to be allocated when the customer order is accepted. The allocation of components may be phased in time if, for example, the promised delivery lead time is longer than the lead time required for product assembly.

For each customer order, it is also important to indicate whether it belongs to the forecasted allowance. When it is included in the forecast, then the remaining allowance needs to be reduced by the size of the order. This is referred to as forecast consumption in the literature [BERT90].

### **Generic material requirements planning (GMRP)**

Before addressing the subject of customer order acceptance in an ATO environment using a generic bill-of-material, it is useful to provide an introduction to the concept of GMRP.

Material requirements planning (MRP) is normally based upon a planning unit that is equal to one product. With generic material requirements planning, however, the planning unit is a set of products. Each set of products may represent a generic product or a specialty and may consist of 0, 1 or more products. In the remainder of this thesis, this type of set is referred to as a **product set (PS)**.

A table is created for each set. Similar to MRP, the time-phased balance between supply and demand is recorded in this table. For each period, the supply may consist of:

- the quantity of products in stock

- the quantity of one or more scheduled receipts from released work orders or purchase orders
- the quantity of planned work orders or purchase orders that have not yet been released.

The demand may consist of:

- the total independent demand remaining after forecast consumption
- the total dependent demand remaining after forecast consumption
- the quantity from customer orders
- the quantity from allocations.

The above-mentioned units all represent the sum of the units belonging to the individual variants of a single product set. This means that it must be possible to add these variant units together. In other words, the units must either have the same dimensions or be convertible to the same dimensions. This will normally be possible since all of the variants belong to the same generic product.

The GMRP table is similar to the MRP table. A major difference between these tables, however, is the line with received customer orders. This line is required since a customer order may incorporate a finished product that is identified and specified indirectly. This topic will be discussed in more detail in the subsection about GMRP and the product demand.

Similar to using MRP, lead times per planned unit are required. This unit is a product in the case of MRP. The unit is a product set in the case of GMRP. The variants of this type of set will not be significantly different from each other in terms of capacity types and requirements. In view of this, it is generally acceptable to use a single lead time for all of the variants. If this leads to problems because of large variances in the lead times, the set could be split up into multiple subsets. This approach is also possible with respect to one of the other planning parameters: the lot size.

Lot sizes and lead times are, in principle, only important for the lowest level specialties. These parameters are only required immediately preceding the explosion of a GP, similar to the MRP situation. This is when the lowest level specialties are processed.

---

This can be illustrated using our chair example. The planning percentages are shown in Figure 9.4. For the purpose of simplicity, we will assume that all of the products have a lead time of one week. The independent demand for chairs is estimated to be 200 units in period 5. Figure 9.5 shows the GMRP tables for three product sets, namely: the top level specialty for the chair, the red lowest level specialty and the red specialty for the seat.

The table at the top of this figure is the table for the universal top level specialty belonging to the generic chair product. The independent demand for 200 chairs is noted at this level. The lead time is zero in the first GMRP table since the product set is not a lowest level specialty. As previously noted, the lead times are only relevant for the bill-of-material relationships and not for choice relationships.

### **GMRP and multivariants**

Until now, we have seen only minor differences between MRP and GMRP. A major difference is apparent when a product set with multiple variants is exploded. This type of product set is referred to here as a **multivariant** (MV).

This section explains why it is not possible to use the MRP algorithm for a multivariant without making special provisions.

In MRP, a product is exploded into its components in connection with the planned release of orders for a product. The planned release of orders occurs after applying the netting logic where the net requirement is calculated by adding the cumulative scheduled receipts to the current stock level and subtracting the cumulative gross requirement. A simple approach to material planning would be to apply this same algorithm to an MV. Since an MV represents multiple variants, however, the release of planned orders in this way would produce a random mix of the variants and would normally not be the same as the sum of the planned orders for the specific variants actually required. When exploding the planned order release for an MV, the planning percentages are used to distribute the releases over the specialties of its generic component.

Gen. product <b>5612G</b> safety stock = 0    allocations = 0    lot size = 1 <b>chair</b> Specialty <b>4</b> (top level)    stock = 0    lead time = 0							
	0	1	2	3	4	5	6
gross requirement						200	
customer orders							
scheduled receipts							
net requirement						200	
planned orders						200	
planned order release						200	
Gen. product <b>5612G</b> safety stock = 0    allocations = 0    lot size = 1 <b>chair</b> Specialty <b>1</b> (red)    stock = 0    lead time = 1							
	0	1	2	3	4	5	6
gross requirement						120	
customer orders							
scheduled receipts							
net requirement						120	
planned orders						120	
planned order release					120		
Gen. product <b>7419G</b> safety stock = 0    allocations = 0    lot size = 1 <b>seat</b> Specialty <b>1</b> (red)    stock = 0    lead time = 1							
	0	1	2	3	4	5	6
gross requirement					120		
customer orders							
scheduled receipts							
net requirement					120		
planned orders					120		
planned order release				120			

Figure 9.5: GMRP tables

In the example above, an independent demand of 200 units has been entered. Using the data from Figure 9.4, the derived requirement for underframes without wheels becomes 144 units, of which 72 are not swivelling. If we assume that 60 units of non-swivelling underframes without wheels are already in stock, then there is a net requirement of 12 units for this specific type of underframe. In order to produce the additional 12 units, 12 non-swivelling stands without wheels will be required. On the other hand, if we had just applied MRP logic to this situation for the MV underframe without wheels, then the net requirement would have turned out to be 42 units based upon 50% of the net MV requirement (= 144 - 60). This is all shown in Figure 9.6.

	Derived requirement	Stock UNDERFRAME without wheels non-swivelling	Net requirement	MRP Net requirement
7453G-2 UNDERFRAME without wheels	144	60	84	84
7291G-1 STAND without wheels swivelling	72	60	12	42
7291G-2 STAND without wheels non-swivelling	72		72	42

Figure 9.6: Comparison of net requirement calculations

A further aspect can be illustrated using the data in Figure 9.6. If we assume that there are now 80 units in stock instead of just 60 units, then a sufficient quantity of this underframe variant is already available in stock. In other words, the net requirement for this variant is zero. The total net requirement for underframes without wheels is the same as the net requirement for non-swivelling underframes:

72 units. Calculating the net requirement according to MRP rules would show a net requirement of 64 units, however. This is equal to the total gross requirement of 144 less the total of 80 units in stock.

If the MRP algorithm were to be applied blindly to the MV, then the example above demonstrates that the following conclusions can be drawn regarding MV's:

- The net requirement for an MV is not necessarily equal to the sum of all of the net requirements for its variants.
- The gross requirement for a generic component cannot be determined by calculating the net requirement for the parent when this is a multivariant.

In order to arrive at a correct determination of the net requirement for a variant, then we must determine the difference between the total downstream demand and the total downstream supply of that variant in the flow of materials. This refers to the stock point of the variant and the stock point of the finished products in which this variant is used. In this way, we are taking an **integral** approach to the product and viewing it throughout the total flow of materials, independent of any product in which the variant may be used.

### **GMRP and Line Requirements Planning (LRP)**

As seen above, an integral approach is required for determining the gross requirements for multivariants. In addition to multivariants, any univariants that may influence the variation within MV's should also be viewed integrally. The stand variants in the example in Figure 9.6 are key determinants for the variation in the underframe. In other words, all of the multivariants and all of the univariants included in a decision tree need to be planned integrally. This means that any specialties involved in the indirect identification of a product must be planned integrally. These specialties are shaded in the chair example presented in Figure 9.4.

One of the planning techniques that uses this type of integral approach is called **line requirements planning (LRP)** [DONS87][DONS89]. This technique is used here whenever an integral approach is required.

---

With this particular planning technique, the quantities such as total stock, allocations and scheduled receipts are determined integrally as well as the gross requirement.

MRP can be used to plan product sets that do not require an integral approach. This is, of course, not absolutely necessary. LRP can also be used for planning these sets. These PS's will always be planned with MRP within the context of GMRP since these PS's are not for an indirect identification.

If a change from an LRP-planned PS to an MRP-planned PS occurs somewhere in the chain, then subsequent PS's after that point in the chain will all need to be planned using MRP. An LRP-planned PS such as this will always be a multivariant. As shown above, the net requirement for this multivariant is equal to the requirement for the total system and may not be equal to the sum of all of the net individual requirements. The net individual requirements in this context represent the requirements of the individual variant that belongs to the relevant multivariant. If the scheduled receipts and stock levels are in balance, then the net requirements will be the same as the sum of the net individual requirements. "In balance" means that the flow of goods initiated by the individual variants is more or less in conformance with the planning percentages. In other words, all of the individual variants of an MV have the same run out time [AXSÄ86]. The run out time is defined here as being the number of periods needed to fulfill the demand based upon the initial stock levels plus the scheduled stock receipts less the allocated stock.

If the flow of goods initiated by a given LRP-planned PS called A is not in balance, then it is possible that there will be stock shortages sooner or later with respect to the MRP-planned components needed to make one or more variants of A. This problem is discussed in more detail in Chapter 11.

### **GMRP and the product demand**

Following this introduction on the generic MRP, we will discuss how GMRP deals with the second core problem that occurs when changing from an MTS environment to an ATO environment. This core problem concerns how production control interacts with the demand for the product. The GMRP is able to provide

support only for the material part of this problem. The variant specified by a given customer order can be identified in three different ways when a generic bill-of-material is used. As previously described, the three ways of identifying a product are directly, directly with parameters and indirectly. Subsequently, we will explore how the GMRP may support the customer order acceptance process when a variant is identified in each of these different ways.

To start with, we can consider customer orders that have a variant which is identified directly. An example of this is the seat frame that is a part of our well-known chair. When a new customer order is accepted, this is added to the customer orders line in the GMRP table. If this order is included in the forecast, the remaining forecast needs to be reduced accordingly ("**forecast consumption**"). This means that an indication of whether an order has been forecasted must be additionally specified for each order. If the demand turns out to be greater than the forecast in a given period, then the remaining forecast for the preceding period should be reduced. If the preceding period's forecast is totally consumed, then any remaining forecast for the period before that should be reduced [LAND89]. The gross requirements line in the GMRP table indicates the remaining, unconsumed forecast. The sum of the gross requirements line and the customer orders line is, thus, equal to the total demand in a GMRP table.

When a delivery date is assigned to a customer order, it is necessary to check whether sufficient product is available or will be available to make the delivery. A so-called "available-to-promise" (ATP) calculation is often used to check this. An example of such a calculation is presented in Figure 9.7. The ATP calculation is based solely upon the available stock, the allocation, the scheduled receipts to stock and the customer orders that have been accepted. This means that the forecast and the normal safety stock level are not included as part of the requirement in calculating this.

The result of this ATP calculation can be interpreted as follows: an ATP of 8 units in period 1 means that a customer order for a maximum of 8 units may be accepted after period 1. After period 3, an order for a maximum of 18 (= 8 + 10) units may be accepted.



Gen. product <b>3265G</b> safety stock = 5    allocation = 7    lot size = 1							
<b>seat frame</b>							
Specialty	stock level = 20				lead time = 1		
	0	1	2	3	4	5	6
gross requirement		10	15		10		
customer orders			5				
scheduled receipts				10			
stock turnover	13	13	8	18	18	18	18
cumulative ATP		8	8	18	18	18	18
ATP		8		10			

Figure 9.7: "Available-to-promise" calculation

The second type of variant found in a generic bill-of-material system is the type that is identified directly with parameters. With respect to the GMRP ,there is no difference between the first and second type of variant in terms of how the forecast consumption and the available-to-promise calculations are made. This is due to the fact that each variant has its own GMRP table.

The third type of variant is the type with an indirect identification. This type of variant requires a different method than above for calculating the forecast consumption and the available-to-promise. The ATP calculation in this case must be based upon data associated with the individual variants. This data is not included in a GMRP table. In fact, one or more of the tables associated with an indirect identification will refer to multivariants. The chair variant presented in Figure 9.8 can be used to illustrate this.

There are two lowest level specialties in this example that are associated with the indirect identification and refer to multiple variants. These two specialties are the red chair and the underframe without wheels that consist of, respectively, seven and two variants. As seen previously in Chapter 8, data can also be recorded for each individual variant. The ATP calculation should be carried out based upon this data.

	lowest level specialty	description		quantity
1	5612G-1	CHAIR	red	10
	7453G-2	UNDERFRAME	without wheels	10
	7291G-1	STAND	without wheels	10
4	7597G-1	ARMREST	swivel with armrests	20

Figure 9.8: Identification of a customer order for a chair variant

A preliminary customer order acceptance test can be carried out based upon the data reported in the relevant GMRP tables. An ATP for each of the relevant GMRP tables can be calculated. A GMRP table is deemed to be relevant when it is associated with a lowest level specialty that is used in the indirect identification of the variant. A customer order could then be rejected based upon these ATP calculations. If an order is not rejected, however, this should not imply that the order definitely will be accepted. A detailed ATP calculation will need to be carried out before this can be determined. This means that passing the preliminary customer order acceptance test should be seen as a necessary but not sufficient condition for customer order acceptance. This test is meant to provide a quick way of determining whether a customer order could qualify for acceptance.

Since the planning for the flow of materials is based upon the lowest level specialties, the forecast consumption of one customer order must be calculated in the case of indirect identification for all of the lowest level specialties involved.

Depending upon the location of the customer order decoupling point, the acceptance test should be carried out before assembly of the finished product or else before the production of the components.

The components and the finished product in an ATO environment need to be allocated in the appropriate time periods upon acceptance of a customer order. This can be done immediately based upon the GMRP tables when the products are identified directly. When the products are identified indirectly, however, this must be carried out for the variants as well as for the lower level specialties that are involved in identifying the variants. The lead times and the bill-of-material

---

quantities for the GP's associated with the relevant lowest level specialties must be taken into account when the allocations are made.

The next question is how to deal with quotations. A quotation can be seen as something between a forecast and a customer order. The components appearing in a quotation may be allocated. If there are too many uncertain quotations, however, these quotations could block the acceptance of future customer orders. In order to resolve this dilemma, Higgins [HIGG92] recommends the use of a time fence. No quotations for work to be carried out before the time fence should remain in the system. We will refer to this time fence as the **quotation time fence**. This time fence should be set, minimally, to be equal to the customer order delivery lead time starting from the beginning of the planning horizon. Similar to the situation with customer orders, a distinction is made between forecasted and non-forecasted quantities for the quotations.

### **GMRP and the configuration process**

As previously discussed, the purpose of the configuration process is to specify the variant within the scope and limitations of a product family which still meets the customer's requirements. The configuration process can be extended if necessary. This process may be limited in terms of the availability of the required components as well as technical feasibility. In other words, the choice of a variant may depend upon logistic considerations as well as technical factors. Logistic limitations can be taken into account through the use of GMRP data in the configuration process. This means that the delivery lead time may also influence the ultimate choice of product variant.

It is not necessary to check the availability of all of the components as part of the configuration process. The cheaper components and low cost parts do not need to be controlled in this way since they are usually abundantly available and will not affect the delivery lead time. Other types of components that may not need to be controlled are the common components within a product family. When the demand for a given product family, in its entirety, has a relatively stable character, then sufficient common components will normally be available at the customer order decoupling point. Components that satisfy one or more of the following criteria

generally need special attention in connection with planning [HIGG92] and, thus, also during the configuration process. These criteria are:

- unique
- long delivery lead times
- expensive
- bulky.

When a component of a potential variant is unique, it is not likely that this component will be kept in stock. If this type of component is then required to produce a specific variant, it will always be necessary to check whether the component is in stock or needs to be ordered.

If components with long delivery lead times are required to produce a variant that is not normally kept in stock, then the situation of whether these components are currently available in stock will likely determine the delivery date for the variant. When components are expensive or bulky, only a limited quantity will normally be kept in stock. This means that the delivery of these components may also lie on the critical path that ultimately determines the delivery lead time for the variant.

The delivery lead time for a customer order is often long when the product is complex. The delivery lead time includes the time necessary to prepare the product specification as well as the time needed to obtain the components and to assemble the product. This may be a lengthy process. The preparation of the product specification could be carried out simultaneously, to some extent, with the requisition of components and assembly of the product in order to shorten the total delivery lead time. This means that the components to be purchased or assembled first, must be fully specified to start with. Thus, the parameter values that determine these specifications need to be known first. As a result, the lead times and delivery times used for GMRP dictate the order in which the parameters need to be specified during the configuration process.

### **3 Assembling the customer orders**

After the customer orders have been accepted and the required allocations have been made, the assembly process needs to be initiated. Work orders are generally issued to achieve this. A work order may incorporate one or more customer orders. Combining several customer orders in this way has the advantage of reducing the

---

total number of work orders on the shop floor. A pick list needs to be generated for the components required to produce each work order. The pick list can be generated from the generic bill-of-material. The required components also need to be allocated. The pick list and the allocations do not need to cover all of the components, however, since the low cost parts and any common components may not need to be controlled by the system.

In the case of complex products it may be necessary to prepare assembly instructions for each work order. In addition, an assembly schedule will need to be drawn up.

Components can be subtracted from stock in two different ways. The first way is to reduce stock levels as soon as the components are physically removed from stock. The second method is to reduce stock levels when the assembly work order is completed. This second method is referred to as back-flushing. Either of these methods can be based upon a generic bill-of-material.

The finished product inventory is increased whenever an assembly work order is completed. In some cases this work order will consist of multiple customer orders. Since each customer order refers to a product variant, any given work order may be comprised of multiple variants. These variants will normally be identified indirectly. The stock levels must be updated for the individual customer orders as well as in the GMRP tables for the relevant multivariants.

When customer orders are shipped, the stock levels are again updated for the individual customer orders and the multivariants. In addition, the allocations for the customer orders in the GMRP tables are released.

The above-mentioned transactions are summarized below in Figures 9.9a and 9.9b. Additional data is updated at the component level when the finished product variant is identified indirectly.

FINISHED PRODUCT LEVEL						
activity	GMRP table			customer order		
	customer order	scheduled receipts	stock	allocation	scheduled receipts	stock
customer order entry	plus			plus		
work order release		plus			plus	
requisition of parts						
work order completion		minus	plus		minus	plus
customer order shipment	minus		minus	minus		minus

Figure 9.9a: Transactions in connection with processing a customer order

### Uncommitted forecast

An uncommitted forecast may exist at the beginning of the planning horizon. This forecast is used to initiate the flow of materials for the components at the finished product level. Once that this has been effected, this forecast no longer serves a purpose and can be deleted. An alternative approach is to introduce a time fence [HIGG92]. In this way, all of the forecasts for periods preceding this time fence can be eliminated from the calculations of the net requirements and the explosion of the gross requirements using the LRP algorithm. This alternative approach has the advantage that the forecasts at the beginning of the planning horizon need to be changed whenever the planning horizon changes. In addition, it is possible to see what the previous forecast was in each case.

The forecast may apply to a specific variant as well as to a multivariant.

**Release of work order components for multivariants**

The manufacture or purchase of the components needed to produce future customer orders is planned using GMRP. Multivariants may be used at the component level. GMRP schedules multiple orders in the case of an MV.

COMPONENT LEVEL							
activity	GMRP table				variant		
	customer order	scheduled receipts	allocations	stock	allocations	scheduled receipts	stock
customer order entry	plus (indirect)						
assembly work order release		plus (indirect)	plus		plus		
shipment of parts			minus	minus	minus		minus
assembly work order completion		minus (indirect)		plus (indirect)			
customer order shipment	minus (indirect)			(indirect)			

Figure 9.9b: Transactions in connection with processing a customer order

This type of order includes multiple variants and is, thus, actually an aggregation of several work orders. How large does a work order for a single variant need to be? Or, in other words, how should the scheduled quantity be distributed across the respective variants? Axsäter uses an approach in which the planned quantity is distributed such that each variant involved has a certain minimum run-out time [AXSÄ86]. This **run-out time** is defined as being the number of periods needed to cover the gross requirements (including the allocations) using the current stock plus the work orders less the allocations less the safety stock. The common run-out

time can be calculated simply based upon the data associated with the multivariant [KARM81]. The size of each work order can then be determined using this run-out time.

#### 4 Uncertainty of demand for finished product variants

Several techniques can be used to deal with the uncertainty of demand in an ATO environment [WEMM84][DONS89]. Most of these techniques make use of a buffer stock of components at the CODP. The advantages and disadvantages of using various techniques will not be covered here. Instead, the discussion here will be limited to summarizing the techniques that are supported by GMRP.

Products that are controlled by an MRP algorithm can normally make use of safety stock and/or safety time to allow for the uncertainty of demand. In this case, all of the products will be univariants.

Products that are controlled by an LRP algorithm can also make use of these two techniques. Nevertheless, this will be different from the MRP situation since LRP uses an integral point-of-view to determine the required safety stock norm [DONS89]. An integral point-of-view means that the total chain is taken into consideration: from the product to the finished product in which this product is used, directly or indirectly. All of the multivariants are controlled using LRP in the case of GMRP. This means that an integral safety stock norm is used for an aggregate of multiple variants.

The option **overplanning** can be used in this case since planning percentages are used in connection with GMRP. GMRP provides an extra feature in this way by providing an opportunity for overplanning. A choice can be made between two alternatives for planning common components when all of the variants of a generic product contain one or more common components and the overplanning option is used. The alternatives are planning these components either with or without overplanning. The use of overplanning means that all of the parent variants are made to stock.

To illustrate this, a chain from the example presented in Figure 9.4 can be examined. For example, we can use the chain consisting of the chair, the seat frame



and the upholstery. An overplanning situation can be created by changing the planning percentage from 60% to 70% for the red chairs and from 30% to 40% for the gray chairs. Assume that the percentage for the blue chairs is left unchanged at 10%. This results in an overplanning of 20%.

	pl. %	stock		CODP seat	CODP seat frame/ upholstery
red chair	70%		gross req.	105	105
			net req.	105	105
gray chair	40%		gross req.	60	60
			net req.	60	60
blue chair	10%		gross req.	15	15
			net req.	15	15
CHAIR			gross req.	150	150
			net req.	150	150
red seat		40	gross req.	105	105
			net req.	65	65
gray seat			gross req.	60	60
			net req.	60	60
blue seat			gross req.	15	15
			net req.	15	15
SEAT		40	gross req.	<b>180</b>	150
			net req.	<b>140</b>	110
SEAT FRAME			gross req.	140	110
			net req.	140	110
red upholstery			gross req.	65	65
			net req.	65	65
gray upholstery			gross req.	60	60
			net req.	60	60
blue upholstery			gross req.	15	15
			net req.	15	15
UPHOLSTERY			gross req.	140	<b>140</b>
			net req.	140	<b>140</b>

Figure 9.10: Overplanning and common components

We will discuss two different situations. In Situation I, the customer order decoupling point is at the seat. This means that the seats will be made to stock. In Situation II, the CODP is at the lowest level in the bill-of-material, namely, at the seat frame and upholstery level. We will assume that 40 red seats are already in stock. At the highest level in the bill-of-material, a gross requirement of 150 chairs is specified.

The gross requirements for the three lowest level specialties are calculated using the planning percentages and the specified total requirement of 150 chairs (Figure 9.10). Since there are no chairs in stock, the net requirement is equal to the gross requirement. The gross and net requirement for the generic CHAIR product is first calculated without providing for overplanning.

The seat specialties are planned using MRP since the seat is not involved in the identification of the chair. Assuming that a lot size of one is used in producing chairs, the gross requirements of the seat specialties are equal to the net requirements of the respective chair specialties.

In Situation I (where the CODP is at the seat), the gross and net requirements of the SEAT GP are equal to the sum of the respective requirements of its variants.

However, in Situation II (where the CODP is at the lowest level), the gross requirement of the SEAT is determined based upon the explosion of the net requirement for the chair. The net requirement of the SEAT is calculated in the normal manner using MRP.

The common component of the seat variants, the seat frame, is determined based upon the explosion of the net requirement of the SEAT GP. This results in 140 units in Situation I and 110 units in Situation II. When the CODP is at the seat, more seat frames will be required than in the situation in which the CODP is at the seat frame and the upholstery.

---

### **Generic MRP in a mixed environment**

It is difficult to find an environment in which only ATO production is used. This is because there are often certain variants within a product family for which there is a high demand or that need to be delivered directly from stock. These variants are kept in stock and assembled prior to the acceptance of customer orders. These products are often referred to as **fast-movers**. The other variants within a product family are the **slow-movers** and are typically assembled to customer order. This situation represents a mixed ATO and MTS production environment.

The possibility of specifying a time-series of gross requirements for each (fast-mover) variant must exist in order to support this type of mixed environment within GMRP. The gross requirement at the product family level consists of several parts. One part is the sum of the gross requirements of the fast-movers. The other part represents the total gross requirement of the slow-movers. Only this part uses the planning percentages specified in the generic bill-of-material.

The gross requirements of the fast-movers that are identified indirectly are also recorded here with the various lowest level specialties involved. The lead times and the quantities per must be taken into account.

### **Summary**

The second use for a generic bill-of-material, the generic MRP, has been described in this chapter. We have seen how the GMRP uses algorithms borrowed from MRP and LRP. GMRP plans finished products as well as component level products as much as possible in aggregate terms. The flow of materials is initiated as much as possible based upon the actual variants. This implies that a number of aspects need to be tracked at the variant level as well as at an aggregate level. These aspects are the stock levels, safety stock levels, allocations, work orders and customer orders. The gross requirements of the variants to be supplied from stock need to be specified per variant whenever variants belonging to a given product family are supplied from stock as well as to customer order.

## 10 Comparison with other product models

---

### Introduction

In this chapter, a comparison is made between the product model developed here (the generic bill-of-material) and the six product models described by Van Veen [VEEN91]. This comparison includes an analysis of the extent to which the product models are suitable for use as a basis for material requirements planning. The following models are discussed:

- 1 a modular bill-of-material [ORLI72]
- 2 the product structure card of Carruthers [CARR76]
- 3 the logical graph of Wedekind and Müller [WEDE81]
- 4 the variant bill-of-material of Schönsleben [SCHÖ85]
- 5 the generic bill-of-material of Van Veen [VEEN91]
- 6 an expert system such as XCON [BARK89] [LEON87].

Using our chair example (see Figure 9.4), we will describe what is included in each of these models. In addition, we will use this example to point out the differences between these models and the generic bill-of-material model developed here. We will use the requirements specification for the product model as presented in Chapter 3 to draw comparisons with each of these models.

It was established that a product model should support:

1. the definition of product families at all levels of the hierarchical product structure
2. the specification of product properties for each product family
3. the specification of planning percentages for each property or combination of properties
4. the definition of individual products within a single product family
5. a specification of which combinations of product properties are valid
6. the translation of one set of product properties into a different set of product properties.

The basic conditions for carrying out material requirements planning were:

1. generation of a material requirements plan based upon the definitions of finished products, components and product families
2. generation of a material requirements plan, if necessary, for a product variant within a product family

3. ability to use planning percentages per property or combination of properties
4. ability to generate plans based upon aggregate information
5. ability to release individual work orders based upon aggregate work orders
6. ability to drive the final stage(s) of production of a finished product based upon customer orders as well as stock replenishment orders.

It is evident from these basic conditions for carrying out material requirements planning that these conditions do not imply that any additional conditions need to be imposed on the product model.

### Modular bill-of-material

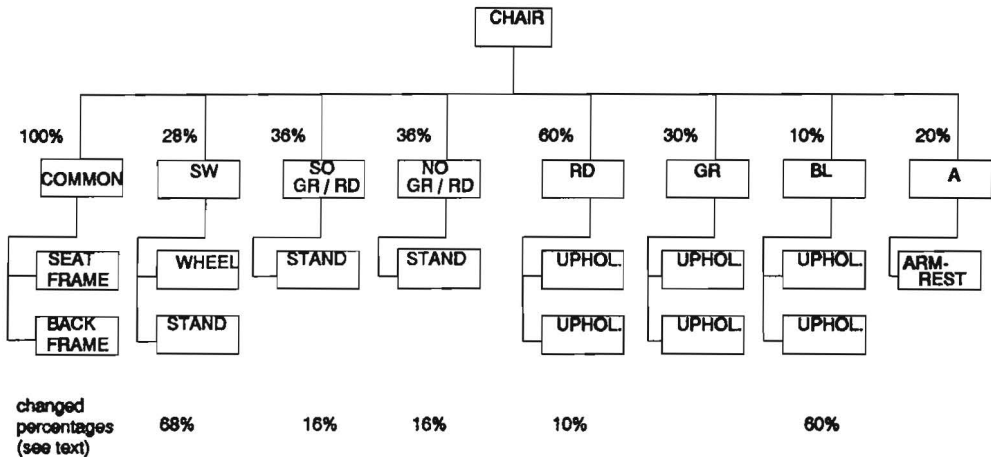


Figure 10.1: Modular bill-of-material for the chair example

We described previously in Chapter 9 how a chair could be modeled in the form of a modular bill-of-material. The modular bill-of-material for the example of a chair is presented in Figure 10.1. The modular bill-of-material is actually not suited for modelling this chair example in view of the dependencies between the parameters (see Chapter 2).

When we compare a modular bill-of-material to the modelling requirements, it is immediately apparent that the modular bill-of-material does not provide for product families at the component level. A modular bill-of-material allows product families

---

only at the finished product level. As a result, it is not possible to simply generate a manufacturing bill-of-material as we have seen in the previous chapter.

Planning percentages can be specified in this bill-of-material, however, it may be difficult to update these percentages in the case of complex product families. This is because the components of complex product families are often dependent upon multiple product properties. This means that many modules need to be created and a specific property may then be associated with multiple modules. If the planning percentage for a certain property is changed, then the corresponding planning percentage will also need to be changed in all of the modules that have this property.

For example, if the planning percentages for the red and blue colors are switched, then five different percentages will be affected in our example. The changed percentages are indicated in Figure 10.1 on the last line. This change can be made in the case of a generic bill-of-material by simply changing the percentages of the two colors involved.

The fourth requirement refers to a mixed MTS/ATO environment. As we have seen in Chapter 9, there are fast-movers in this type of environment that can also use common components found in the rest of the product family. There is no facility in a modular bill-of-material for keeping track of these fast-movers within a given product family. On the other hand, it is always possible to create a separate bill-of-material for each fast-mover. This situation creates a number of problems for material planning, however. These problems are described by Sari in the form of a case study [SARI81].

A modular bill-of-material cannot satisfy the fifth requirement that deals with documenting the valid combinations of product properties. This can be illustrated using the example of a blue chair without wheels. The fact that this is meant to be an invalid product configuration cannot be recognized from the information in the modular bill-of-material. For this reason, Van Veen [VEEN91] imposed the additional restriction in his second assumption regarding a modular bill-of-material whereby no dependencies are permitted between the options of different features. When using a generic bill-of-material, on the other hand, it is possible to document restrictions such as this directly.

The modular bill-of-material cannot directly satisfy the sixth requirement (translating certain parameters into other parameters).

### **Product structure chart**

Carruthers [CARR76] has designed a **product structure chart** to accommodate the shortcomings of the modular bill-of-material with respect to the facility for documenting restrictions. The valid and invalid combinations of parameter values for a product family are listed on this chart.

A product structure chart for our chair example is presented in Figure 10.2. The components needed for certain combinations of parameter values are listed in the shaded area. The finished product variant is associated with one of the three basic products, namely: the basic RED, GRAY and BLUE chairs characterized by the basic COLOR parameter. In addition to the basic parameters, Carruthers makes a distinction between two other types of parameters, namely: the variant parameters and the option parameters. These parameters need to be mutually independent from each other. For this reason, the WHEELS and SWIVEL parameters have been combined to a single parameter called WHEELS/SWIVEL. Another way to achieve this data independence would be to increase the number of basic products. This means that the number of basic parameters needs to be increased. We could add a basic parameter called WHEELS in this example to create the following basic product variants: RED/WHEELS, RED/WITHOUT WHEELS, GRAY/WHEELS, GRAY/WITHOUT WHEELS and BLUE/WHEELS.

The option parameter is different from the variant parameter since an option parameter does not need to be determined.

basic parameter	COLOR		RED	GRAY	BLUE
basic product			basic chair red	basic chair gray	basic chair blue
variant parameter	WHEELS/ SWIVEL	YES/ YES	7291 SW 6439 W	7291 SW 6439 W	7291 SW 6439 W
		NO/ YES	7291 SO	7291 SO	invalid
		NO/ NO	7291 NO	7291 NO	invalid
option parameter	ARMREST	YES	7597 A	7597 A	7597 A

Figure 10.2: Product structure chart for the chair example

Since the product structure chart is used in conjunction with a modular bill-of-material, most of the shortcomings of using a modular bill-of-material remain. The one exception is the possibility of documenting the restrictions to some degree.

**Logical graph**

The concept of a **logical graph** as developed by Wedekind and Müller [WEDE81] is an extension of the bill-of-material concept. A bill-of-material can be expressed in terms of a graph in which the nodes represent products and the lines represent bill-of-material relationships. Three other types of nodes can be found in addition to the product nodes in a logical graph:

- 1 the AND node represents a set of lower level nodes, all of which are important during the configuration process
- 2 the XOR node represents a set of lower level nodes for which a choice must be made to select one of these during the configuration process
- 3 an empty node represents a decision point that has no implications for material planning.



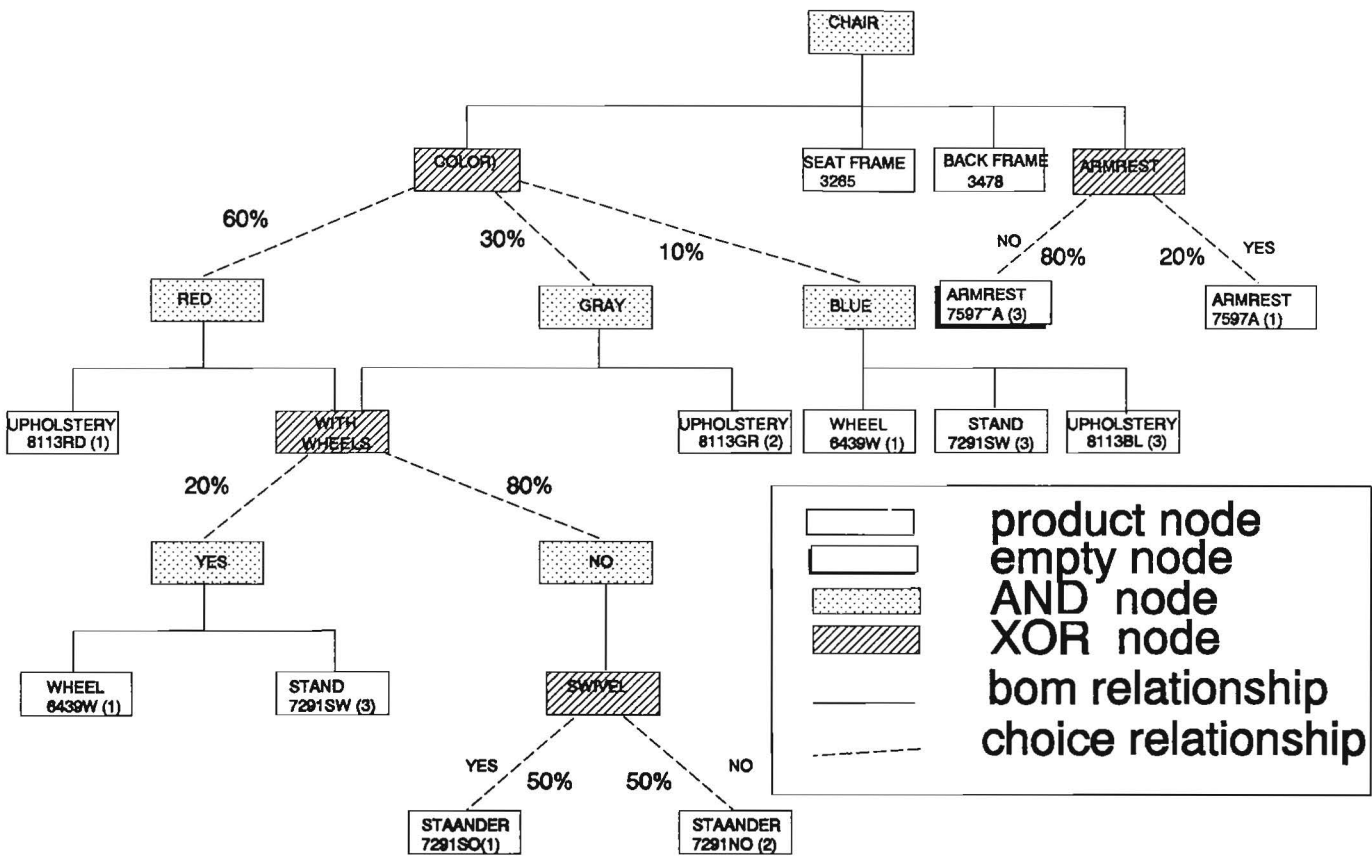


Figure 10.3: The logical graph for the chair example

---

In addition to the bill-of-material relationships, the choice relationships are also represented as lines connecting the nodes in a logical graph. These relationships are indicated by dotted lines in the model of the chair product family presented in Figure 10.3.

To start with, a comparison with the product model requirements shows that a product family can be defined at only one level. Planning percentages can be added to the choice relationships. Restrictions can be included in such a graph, but this is not always easy to do (as in our example). It is not possible to identify an individual product within a product family. On the other hand, however, it is not difficult to translate one set of properties into a different set of properties.

### **Variant bill-of-material**

A variant bill-of-material has been developed by Schönsleben [SCHÖ85]. This type of bill-of-material can be seen as an extension of a normal bill-of-material. The addition is in the bill-of-material relationship. A bill-of-material relationship is seen to be a set of choice relationships in this case. All of these choice relationships have the same parent, but different component variants. Using the selected parameter values, one relationship is chosen from this set of relationships when a variant is configured. This means that a condition is assigned to each relationship within the set. If the condition is satisfied, then the associated relationship is chosen.

A variant at the finished product level is described using parameters that are mutually independent. A choice relationship within a bill-of-material relationship can then be selected based upon the parameter descriptions. The component variant can be determined in this way. The bill-of-material relationships with the choice relationships are found only at the lowest level of the bill-of-material.

The variant bill-of-material for the chair example is presented in Figure 10.4. The choice relationships with the respective condition descriptions are presented in Figure 10.5.

Comparing this with the requirements for the product model shows us that it is possible, in principle, to define product families at all levels in the bill-of-material

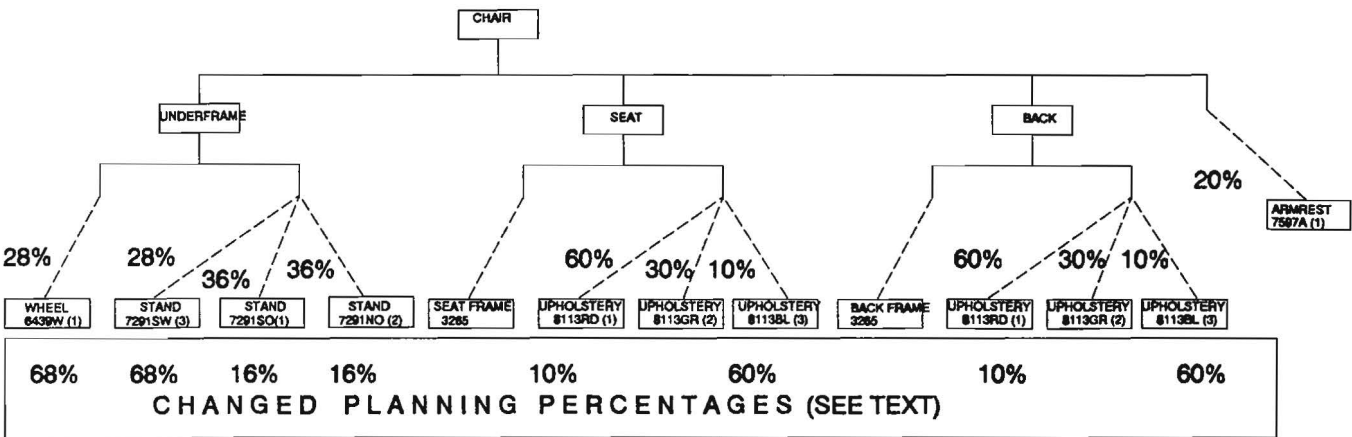


Figure 10.4: The variant bill-of-material for the chair example

except at the lowest level. The planning percentages can be recorded along with the choice relationships. A major effort is required to maintain the data for complex product families using this type of bill-of-material, similar to a modular bill-of-material situation. The required effort is even greater in the case of a variant bill-of-material since a planning percentage is associated with each individual primary product rather than with a module as in the case of a modular bill-of-material. A module normally includes multiple primary products.

This can be illustrated using the same example as with the modular bill-of-material. The original planning percentages for the red and blue colors are swapped in this example. This results in a situation in which eight of the eleven planning percentages need to be changed in the variant bill-of-material for the chair (see Figure 10.4).

Similar to the situation with the modular bill-of-material, the dependencies cannot be recorded here.

parent	component	condition
underframe	wheel	WHEELS=YES
underframe	stand SW	WHEELS=YES and SWIVEL=YES
underframe	stand SO	WHEELS=NO and SWIVEL=YES
		COLOR=RED or GRAY
underframe	stand NO	WHEELS=NO and SWIVEL=NO
		COLOR=RED or GRAY
seat	seat frame	
seat	red upholst.	COLOR=RED
seat	gray upholst.	COLOR=GRAY
seat	blue upholst.	COLOR=BLUE
back	back frame	
back	red upholst.	COLOR=RED
back	gray upholst.	COLOR=GRAY
back	blue upholst.	COLOR=BLUE
chair	armrest	ARMREST=YES

Figure 10.5: Choice relationships with conditions for the variant bill-of-material for the chair example

---

### Van Veen's product model

The generic bill-of-material developed by Van Veen will be referred to here as the "V" product model in order to avoid any confusion with the model developed as the basis of this thesis.

The V model recognizes generic products that are characterized by parameters, similar to the generic bill-of-material situation. In this case, however, the generic products are only identified directly. This means that all of the parameters needed for identification and specification need to be recorded along with the appropriate generic product.

To illustrate this, the chair example based upon the V product model is presented in Figure 10.6. The parameters are noted along with the generic products. This means that the generic CHAIR product has all four of the parameters needed to describe a chair variant. Only the COLOR parameter is present for the generic CHAIR product in our model. This is due to the fact that indirect identification is permitted in the generic bill-of-material. The parameters are in some sense distributed throughout the bill-of-material. In the V product model the parameters are specified as high as possible in the bill-of-material structure. The parameters are specified as low as possible in the generic bill-of-material, however.

The bill-of-material relationship in the V product model is a set of parent-component relationships, called CAS (Cover Aggregation Set) relationships. This relationship always consists of only a single parent-component relationship in the case of a generic bill-of-material. When defining a CAS relationship, it is possible to specify multiple parent-component relationships with different bill-of-material quantities. This can be useful in some situations. The test case presented in Chapter 3, for example, refers to a Desk Island comprised of one to four desks. If the desks are always identical, then the quantity-per in the parent-component relationship representing Desk Island to Desks will vary from 1 to 4. The quantity-per is then a parameter with a value of 1, 2, 3 or 4. It is not possible to model this situation directly based upon a generic bill-of-material. By using phantoms, however, a generic bill-of-material can accommodate this type of situation. This type of situation with multiple bill-of-material quantities does not occur very often, as Van Veen points out. A situation that does occur frequently, however, is when a generic

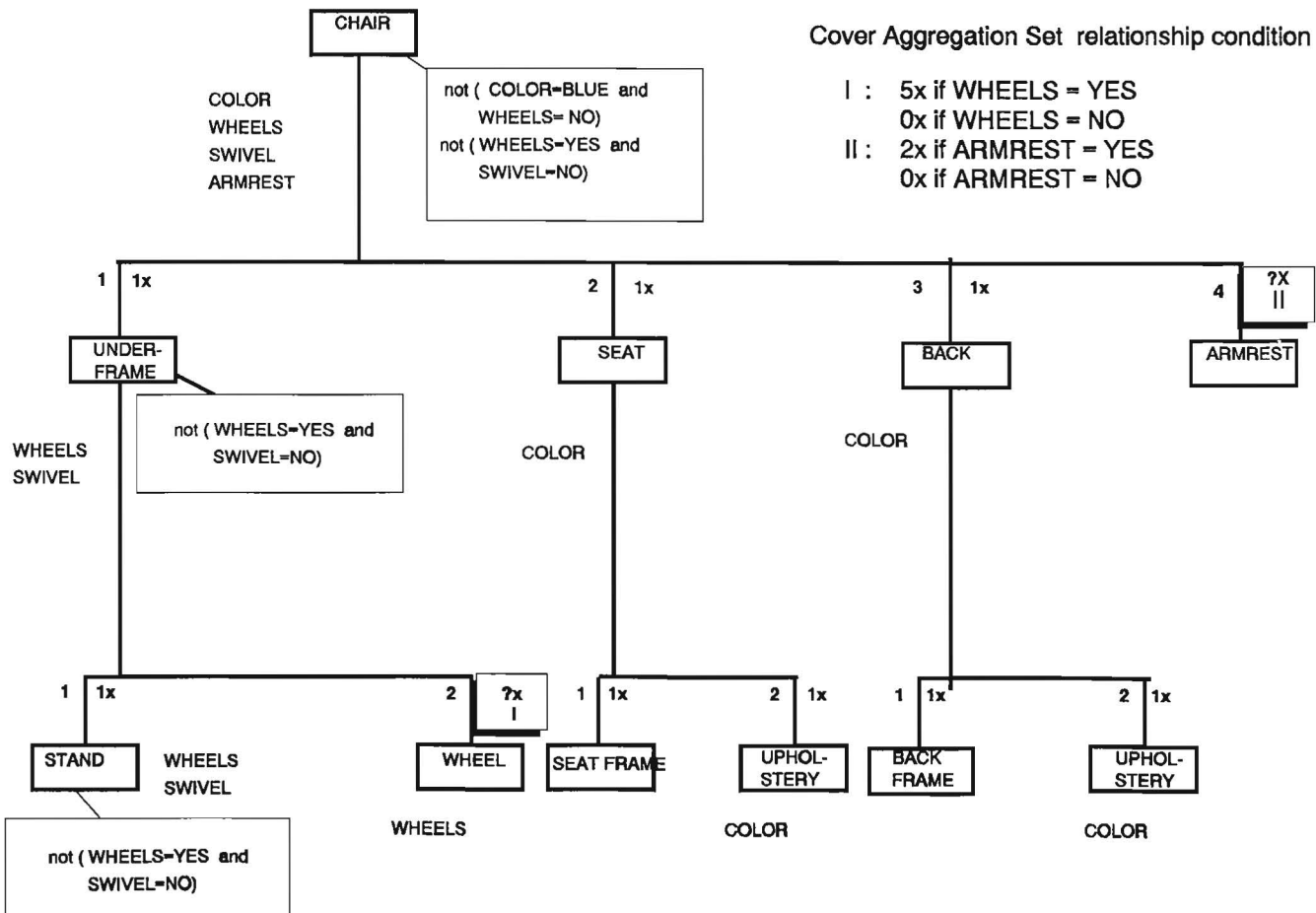


Figure 10.6: The chair example using Van Veen's product model

component is either chosen or skipped in the process of specifying a variant. A generic bill-of-material can accommodate this type of situation easily. Examples of this are the WHEEL and ARMREST components in the chair example.

Three types of conditions can be identified in connection with the V product model. The first type of condition is similar to the generic product situation in which the dependencies between the parameter values of a generic product are described. Figure 10.6 shows a generic CHAIR product for which two conditions have been specified that are required in order to adequately describe the chair. The second condition refers to the parameter values that are also required for the generic Underframe and Stand products. As a result, this condition must be recorded again with these generic products. When a generic bill-of-material is used, however, this condition only needs to be recorded once, namely, with the Stand (Figure 6.2).

The second type of condition involves the parameter values of the CAS relationship and the parameter values of the parent associated with this CAS relationship. Four examples of this type of condition are shown in Figure 10.6. Two of these are associated to the armrest and two are related to the wheel.

The third type of condition involves translating a number of parameters of a parent into the parameters associated with its component. Van Veen refers to this type of condition as a conversion rule that is associated with an element of the CAS relationship. This type of translation is incorporated in the decision conditions for a generic product in the case of a generic bill-of-material. External parameters are translated into internal parameters. The external parameters are inherited from the parent. The internal parameters are, in turn, passed on as necessary to the components at the lower level. The external parameters represent the properties of the interface associated with the generic product. In this way, they describe the environment in which the variants of the generic component are to be used.

When one or more standard interfaces have been defined to cover all of the current and future variants of a generic component X, then this generic component can be used by a new generic parent by simply defining a new bill-of-material relationship. A prerequisite is, of course, that at least one of the interfaces with the new generic parent can be defined using the internal parameters.

---

If this same approach is followed using the V product model, then the translation of one set of parameters into a different set of parameters will need to be recorded in the form of a new set of conversion rules associated with the new CAS relationship. This would be rather tedious to do.

In the case of a generic bill-of-material, in addition to the possibility of recording the translation of parameters within a number of choice conditions, parameters can also be translated using an algorithm.

The V product model satisfies almost all of the product modelling requirements. There are no problems in meeting requirements 1, 2, 5 and 6. The V product model cannot satisfy the requirement of recording planning percentages, however, due to the aspect of having mutually independent parameters.

If this shortcoming of using a V product model is resolved, then an additional problem will surface that will frustrate any efforts of using this model for material planning. This problem is that the V product model does not support subsets within a generic product. Subsets are needed to keep track of a variety of planning data.

### **Expert system**

A frequently used solution to the problem of configuring variants is to make use of an expert system. This approach was used as early as 1979 by the Digital Equipment Corporation [BARK89] to configure computer systems. The expert system in this instance was initially called R1 and was later referred to as XCON. It was written in the OPS5 programming language. This type of system is typically based upon **production rules** to store the knowledge needed to perform a configuration. A production rule consists of a condition part and an action part. The conditions are stored in the condition part. This is used to decide whether the actions stored in the action part are to be performed. A set of such rules is referred to as the **rule base**.

The production rules needed to configure a chair variant are shown in Figure 10.7. A large number of production rules may be required in practice. More than 10,000 rules were used by the XCON configurator in 1988. The number of rules also affects the processing time required to generate a configuration and the amount of



effort needed to maintain such a system. At DEC, several hundred major products are introduced each year. This results in the need to change approximately 40% of the rules in XCON each year. A technique has been developed for decomposing and structuring the rule base in order to be able to cope with this problem of maintaining a reliable configurator system. This technique involves the formation of logical groups of production rules and defining hierarchical relationships. An example of a group in this case could be a subassembly such as a disk drive.

CONDITION PART	ACTION PART
1. If COLOR=RED	then RED SEAT and RED BACK
2. If COLOR=GRAY	then GRAY SEAT and GRAY BACK
3. If COLOR=BLUE	then BLUE SEAT and WHEELS=YES
4. If WHEELS=YES	then SWIVEL=YES
5. If WHEELS=NO	then COLOR=RED or COLOR=GRAY
6. If ARMREST=YES	then ARMREST
7. If WHEELS=YES and SWIVEL=YES	then WHEELS,SWIVEL FRAME
8. If WHEELS=NO and SWIVEL=YES	then NO WHEELS,SWIVEL Underframe
9. If WHEELS=NO and SWIVEL=NO	then NO WHEELS,NO SWIVEL Underframe

Figure 10.7: Rule base for an expert system for the chair example

If we compare the expert system described above with the product modelling requirements, then we can conclude that no entities are defined for product families at either the finished product level or the component level. This means that a variety of planning data and transaction data cannot be recorded here. The planning percentages could perhaps be incorporated in the production rules. An expert system satisfies all of the other requirements.

The majority of expert systems include a user-friendly interface and are extremely suitable for use as configurators. The production rules can be structured and

recorded based upon a generic bill-of-material. Thus, using a combination of an expert system and a generic bill-of-material to create a rule base should provide the basis for a power configurator.

### Summary

In this chapter, a number of different product models have been compared with the product model developed in this thesis. More specifically, an evaluation has been made of the extent to which these models satisfy product modelling requirements that have been formulated. Generally speaking, most of these models can provide valuable support in configuring products, however, a majority of these models are unsuitable for use as the basis for material requirements planning (see Figure 10.8).

	ORLI	CARR	WEDE	SCHÖ	VEEN	XPER
1. product families at all levels	-	-	-	+	+	-
2. product properties	+	+	+	+	+	+
3. planning percentages	+	+	+	+	-	-
4. individual product within product family	-	-	-	-	-	-
5. valid combination of product properties	-	+	+	-	+	+
6. translation product properties	-	-	+	+	+	+

Figure 10.8: Summary of the comparison

## 11 Concluding observations

---

### Introduction

To start with, the prototype system developed to support the research discussed here and based upon the principles covered in the previous chapters, will be described in this chapter. This chapter concludes with a summary of several areas recommended for further research.

### Prototype system

One of the knowledge products of this study is a working prototype system that was built for two reasons to support the research presented here. The first reason was to test whether all of the principles described here were still valid when integrated into a single system. The second reason was to test how well these principles work in practical situations.

The prototype system consists of the following three parts:

- 1 a product model for documenting product families (generic bill-of-material (GBOM))
- 2 a configurator for generating identifiers for variants and a bill-of-material generator for producing bills-of-material for a specified variant
- 3 a system for material requirements planning based upon the product model (generic material requirements planning (GMRP)).

### Product model

The prototype system has been built using the MIRACLE 4GL tool developed by BAAN International. The test case described in Chapter 3 was modelled in accordance with the principles of the generic bill-of-material and subsequently incorporated in the prototype system. The test case was not fully implemented, however. Portions not implemented include the chair and the variable width and height of the desks in the Desk Island. The chair described in the test case is essentially the same chair example used in the previous chapters. The color combinations between the desks and the chairs are not really any different in the product model than in the case of the desk and the joining pieces. The variable width and height could be implemented using continuous parameters and specific algorithms as we have seen in Chapter 6.

Approximately 400,000 variants exist for the Desk Island. This is easy to calculate. Without the table extension and for any given material, color, length, width and height, each desk is available in 8 different variations. These variations are:

- 1 with drawer unit on the left, pen drawer and two normal drawers (drawer unit I)
- 2 with drawer on the left, pen drawer and a hanging folder drawer (drawer unit II)
- 3 with drawer unit I on the right
- 4 with drawer unit II on the right
- 5 with drawer units I on both the left and the right
- 6 with drawer unit I on the left and drawer unit II on the right
- 7 with drawer unit II on the left and drawer unit I on the right
- 8 with drawer unit II on both the left and the right.

With a table extension, six different configurations of the drawer units are possible. Two variations are feasible for the desk part (drawer unit I or II), while one additional variation is possible with respect to the table extension, namely, without drawer unit. The number of variations associated with each desk number are listed in the column "desk number position" in Figure 11.1.

A connecting segment is required when multiple desks are incorporated in a Desk Island. The connecting segment may have a different color than the actual desks. The numbers of color/material combinations are listed in the column "desk/connecting segment." The next column lists the number of possible variations in the shape of the connecting segment. Finally, the last column shows the total number of variants.

In addition to the various Desk Island variants, component variants can also be generated based upon the same information. This results in 30 different variants of the table extension, 42 variants for the connecting segment, 10 variants for the drawer unit and 20 variants for the desk without drawer units.

number of desks	desk number position .x : number of variations in the drawer unit				desk/connecting segment	connecting segment	desk	total
	1	2	3	4	color/material combinations	shape	standard sizes	variant 405,892
1	N 8x	-	-	-	5x	2x(incl.P) 1x	1x 3x	80 120
	L 6x	-	-	-	5x	2x(incl.P) 1x	1x 3x	60 90
	R 6x	-	-	-	5x	2x(incl.P) 1x	1x 3x	60 90
2	N 8x	N 8x	-	-	13x	3x	4x	9,984
	L 6x	R 6x	-	-	13x	3x	4x	5,616
	L 6x	L 6x	-	-	13x	3x	4x	5,616
	R 6x	R 6x	-	-	13x	3x	4x	5,616
3	N 8x	N 8x	N 8x	-	13x	2x	4x	53,248
	L 6x	L 6x	L 6x	-	13x	2x	4x	22,464
	R 6x	R 6x	R 6x	-	13x	2x	4x	22,464
4	N 8x	N 8x	N 8x	N 8x	13x	1x	4x	212,992
	L 6x	L 6x	L 6x	L 6x	13x	1x	4x	67,392

N = no table extension R = table extension on right L = table extension on left  
Figure 11.1: Number of variants for the Desk Island

The number of entities needed to specify the test case are presented in Figure 11.2 [MOUL89]. It is surprising to see that so many entities (15,178) are required to define the relationship between the parameter value and the specialty. This quantity could be reduced by applying the decomposition method for decision trees described in Chapter 6. In this way the decision trees would be split up into similarly structured groups and the number of specialties would be reduced from 2,384 to 1,968. In this way the number of parameter values per specialty and the number of choice relationships would be reduced from, respectively, 15,178 and 1,380 to 10,671 and 59 (see NEW-1 in Figure 11.4).

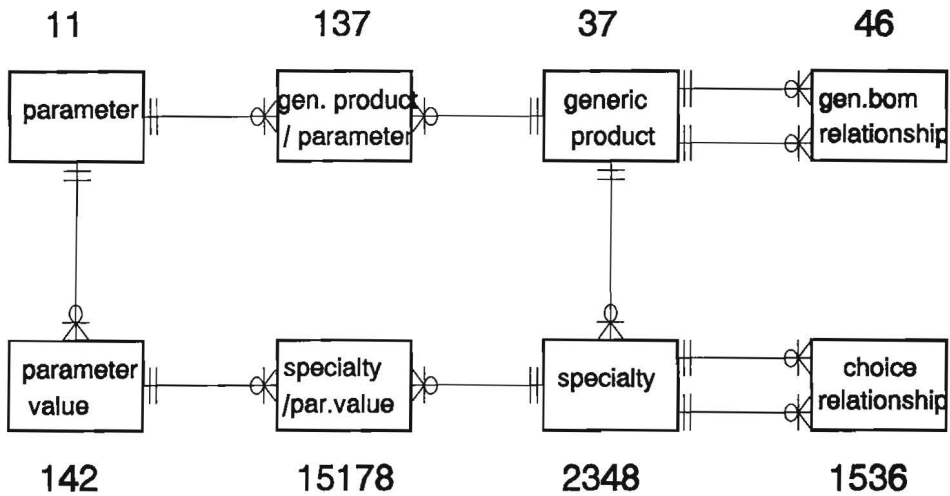


Figure 11.2: Number of entities per entity type for the test case

A variable quantity-per is required in a number of instances in this test case. As seen previously in Chapter 10, this is also the case with respect to the number of desks used in a Desk Island. This number may vary from 1 to 4. Phantom generic products in the prototype system can be used to specify such variable bill-of-material quantities. A phantom generic product represents only phantom products or products that are not kept in stock. These products always have a lead time of zero and a lot size rule of "lot-for-lot." Work orders cannot be issued for these types of products.

The structure shown in Figure 11.3 has been used to resolve the problem of having a variable quantity-per for the number of desks in the Desk Island. Only the

---

QUANTITY parameter is shown in this figure; the other parameters are not mentioned. Four phantom generic products can be identified in this figure, namely, the first, second, third and fourth desks. One specialty is shown for each phantom generic product. This specialty is characterized by one or more parameter values assigned to the QUANTITY parameter. If two desks are specified for a Desk Island (i.e., QUANTITY = 2) using the Desk Island GP, then both the first desk and second desk are included in the variant of the Desk Island to be configured. The desk component is associated with the first desk as well as with the second desk. In this way, the first desk can be configured differently from the second desk.

A large number of the parameter value/specialty relationships (7,344) refer to the specialties of phantoms in this way. It would be useful in this type of situation to define a type of CAS relationship (see Chapter 10) such as that used by Van Veen in his model. As we mentioned in Chapter 10, however, this type of situation does not occur very often in practice.

A further reduction in the number of entities is possible without compromising the generic bill-of-material concept. This can be realized by creating a facility for generating most of the specialties in addition to the existing facility for generating variants. This means that it is possible to wait to create a specialty until it has been decided that it will actually be required. This can be useful when, for example, a specialty is involved in registering the stock of a variant.

Two attributes need to be added to the GENERIC PRODUCT/PARAMETER entity types in order to provide the necessary facility for recording the data needed to generate a specialty. These attributes indicate whether a parameter is internal and/or external in connection with a specific generic product. If all combinations of parameter values are valid, then all of the top level and lowest level specialties can be generated.

When there are one or more invalid combinations of parameter values, then the valid specialties will need to be defined explicitly for each of allowed combinations of parameter values for the parameters involved. An additional attribute needs to be added to the SPECIALTY entity type to be able to make a distinction between specialties that have been specified and specialties that have been generated. The

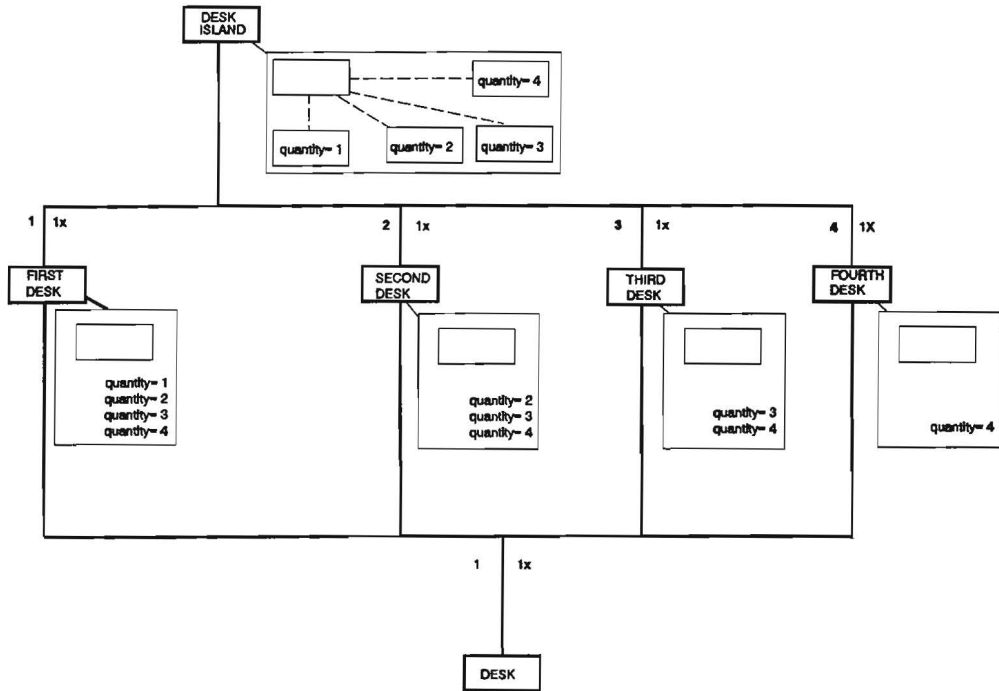


Figure 11.3: Using phantoms to provide for variable bill-of-material quantities  
decomposed version of the decision tree is also used for this purpose.

The variant identification should be recorded in terms of the chosen parameters at the level of the generic products involved in this decision process in order to eliminate the need to make constant use of data stored at the specialty level. This leads to a significant reduction in the number of entities in our test case of the Desk Island.

The number of entities can be reduced by 96.5% in this case. The effort needed to keep this data up-to-date can be drastically reduced in this way. In order to achieve this, a program module must be implemented to generate and delete specialties in all of the situations in which specialties are used.

The generic bill-of-material for the test case is presented in this form in Appendix. The text printed in bold letters describes how the Desk Island is constructed. To start with, the relevant parameters are specified for each generic product and an indication is provided of whether a parameter is internal and/or external.



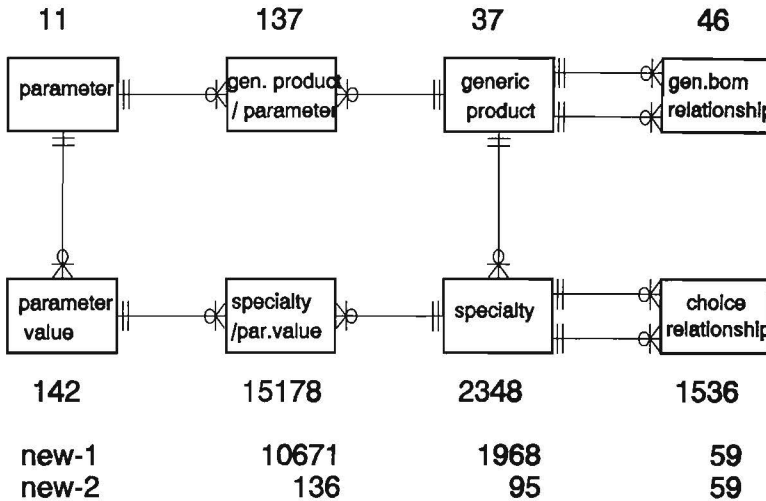


Figure 11.4: Number of entities per entity type for the test case (after reducing the number of entities)

Subsequently, the decision trees for the relevant generic product are shown. Parameter values are assigned to a specialty when it is used for the first time. The decision tree also includes the planning percentages.

The second reason for building a prototype system as part of this study was to test whether the model actually works in practice. The important question was whether it would be possible to model existing product families in a reasonable fashion using the product model.

The following product families have been modelled:

- 1 TV sets (2 families) [DUMO88][YNTE92]
- 2 tent attachments for campers [DOOR92]  
models and specifications for two families for constructing a personal computer system
- 3 sunroofs [JONG91]  
system programmed and implemented using Pascal
- 4 fabrics [LANS92]
- 5 transport castors [LEEN91]
- 6 truck engines [SCHR90]
- 7 variable transport systems [VEN91]

- 8 gas pumps [RED92]
- 9 X-ray equipment [SLOB93]  
system constructed for a personal computer application
- 10 fluorescent lighting (2 families).

It was possible to use the generic bill-of-material to model all of these families. For the most part, these product families were all specified by persons inexperienced in the use of a generic bill-of-material as well as inexperienced in modelling product families. On the average, a period of one month was needed for each family to collect data, specify the family in terms of a generic bill-of-material and documenting the findings. Approximately half of this time was required just to collect the required data.

The numbers of entities are listed in Figure 11.5 for several entity types and product families that were used to model the product families.

	generic products	parameters	parameter values	specialties
TV sets	22	18	49	150
sunroofs	9	25	30 + 14 subranges	34 rules
fabric	32	27	103	236
transport castors	43	23	105	194
variable transport systems	22	4	4 subranges	35 + 9 algorithms
gas pumps	73	50	229	90 (partial)
X-ray equipment	104	40	89	86

Figure 11.5: Number of entities per product family

## **GMRP**

A system for material requirements planning (GMRP) was developed based upon a generic bill-of-material [EEKH93]. This system was tested with a somewhat extended version of the chair, namely, a child's high-chair. A so-called **cover calculation** was included in the GMRP algorithm in order to provide for sufficient planning in a mixed ATO/MTS environment with major differences in the quantities of fast-moving and slow-moving products to be sold. It is possible to be able to satisfy the market demand in a specific period at an aggregate level (i.e., the generic product level) without being able to satisfy any of the demand at a detailed level (the variants) when there are large differences between the quantities of variants sold within a single generic product. In this situation the flow of goods upstream from the generic product will be initiated too late for the products controlled by MRP. The cover calculation is used to resolve this problem.

The cover calculation takes the forecasted demand for each fast-mover and essentially deducts this from the supply for which allocations have been made. This means that the demand based upon a time-series of forecasted gross requirements (after forecast consumption) and customer orders is netted against the supply of stock and issued work orders. By using the cover calculation in this way, the risk is averted of not being able to meet the net requirements created by the generic products at the beginning of the planning horizon. It is necessary to know the net requirements in order to plan the slow-moving components of the relevant GP that are controlled by MRP.

The prototype system will need to be enhanced with a module to issue work orders in order to provide the facility for testing material requirements planning in practice with this prototype. In particular, further study will need to be carried out with respect to fine-tuning the various parameters of the GMRP system. Examples of these parameters are the safety stock levels and lot sizes.

### **Recommendations for further study**

To start with, it will be important to focus future research on evaluating the performance of the GMRP in practical situations and, particularly, using the GMRP in dynamic environments. As indicated previously, the question needs to be

addressed of how the various GMRP parameters should be set. Which criteria should be used for this? The need for further study involves parameters such as safety stock levels and lot sizes. In addition, it will be important to investigate the level at which forecasts need to be made: at the component level, the subassembly level or the final assembly level? Under which circumstances? Which components of a finished product family should be controlled by GMRP and which stock should be controlled? Which variants or specialties within a generic product should be subjected to GMRP and which should not?

The generic bill-of-material concept has been developed particularly for use in an ATO/MTS environment. An important research issue could be to determine to what extent a generic bill-of-material can also be used to support an MTO and an ETO environment. In connection with this, the documentation of general product and component design knowledge should be considered within the context of a generic bill-of-material. This topic was considered to some extent in Chapter 7.

Production planning is an important, labor-intensive function within a company that supports an MTO/ETO environment. A generic routing could be established to support this function more effectively. A generic operation could be used to represent a set of possible operations within this context. Such a generic operation could inherit characteristics from its generic parent as well as generic capacity. A significant amount of the production planning knowledge could be recorded as part of this type of generic routing.

Drawings are important and easy-to-use forms of documentation in an MTO/ETO environment. Drawings consist of a large variety of geometric components. These components are found in many forms and sizes. Size could be coded as one of the parameters. Such parameters could be determined using a configurator based upon a generic bill-of-material. Upon completing the configuration of a variant, it should also be possible to automatically generate a drawing of this variant.

The basis for planning in an MTO/ETO environment is normally a network. Networks of similar products often include the same activities and the same relationships between these activities. This suggests that it would be worthwhile to investigate whether such networks could be seen as generic networks.

Further study is also recommended to determine to what extent a generic bill-of-material can provide a suitable basis for generating software. If this is practical, then it should be possible to configure the software as well as the hardware, simultaneously, for a specific system.

The above-mentioned suggestions for further research deal with the question of viewing a number of data structures from a generic point-of-view. This implies that it may be feasible to extend a relational database to include standard features to support certain types of generic entities.

### **Conclusion**

The purpose of this study was to improve the information systems to support production control within industrial companies with complex products that are supplied as a number of variants.

Part I of this thesis covered the production control issues faced by companies with many variants. One of the conclusions is that the most important modules of an information system in this type of environment are the modules used to document the product, the configurator used to identify a variant and the material requirements planning module. Another conclusion is that the information system in this type of environment must be able to work with aggregated information, particularly the master production scheduling, material coordination and work order issue functions. A requirements specification for a product model was developed to conclude Part I. These requirements also included the two most important applications within the product model to be developed, namely, the configurator and the material requirements planning applications.

A product model for product families, called the generic bill-of-material, was developed in Part II. The basic concept of a generic bill-of-material is the fact that a finished product variant in an ATO/MTS environment is determined by the component variants at the lowest level in the product structure.

Two designs for applications based upon a generic bill-of-material were presented in Part III. One application was the configurator needed to support the identification and specification of a product variant. The other application was the generic

material requirements planning system (GMRP). It was seen how GMRP focuses on aggregate planning at the finished product level as well as at the component level.

---

**APPENDIX: GENERIC BILL-OF-MATERIAL FOR THE TEST CASE**

LEVEL bill-of-material	LEVEL decis.tree	PRODUCT description number	QPER PERC
<b>1</b>		<b>1037 Desk Island</b>	
<b>PARAMETERS</b>		length	I
I: internal		width	I
E: external		height	I
		material	I
		color	I
		quantity	I
		position of table extension	I
	1	sp1 UNIVERSAL	
	*1	sp2 STANDARD SIZE 1	25%
		length	180
		width	80
		height	90
	*1	sp3 STANDARD SIZE 2	25%
		length	180
		width	90
		height	90
	*1	sp4 STANDARD SIZE 3	25%
		length	200
		width	80
		height	90
	*1	sp5 STANDARD SIZE 3	25%
		length	200
		width	90
		height	90
	1	sp6 UNIVERSAL	
	*1	sp7 WOOD	50%
		material	wood
	*2	sp8 PLASTIC	50%
		material	plas
	**1	sp9 LIGHT GRAY	25%
		color	lgra
	**2	sp10 GRAY	25%
		color	gray
	**3	sp11 BLACK	25%
		color	black
	**4	sp12 BROWN	25%

		color	brown
	1	sp13 UNIVERSAL	
	*1	sp14 ONE DESK	25%
		quantity	1
	**1	sp15 NO TABLE EXTENSION	33%
		pos. of table ext.	none
	**2	sp16 LEFT TABLE EXT.	33%
		pos. of table ext.	left
	**3	sp17 RIGHT TABLE EXT.	33%
		pos. of table ext.	right
	*2	sp18 TWO DESKS	25%
		quantity	2
	**1	sp15 NO TABLE EXT.	25%
	**2	sp16 LEFT TABLE EXT.	25%
	**3	sp17 RIGHT TABLE EXT.	25%
	**4	sp19 RIGHT+LEFT	25%
		pos. of table ext.	right
	*1	sp20 THREE DESKS	25%
		quantity	3
	**1	sp15 NO TABLE EXT.	33%
	**2	sp16 LEFT TABLE EXT.	33%
	**3	sp17 RIGHT TABLE EXT.	33%
	*1	sp21 FOUR DESKS	25%
		quantity	4
	**1	sp15 NO TABLE EXT.	50%
	**2	sp16 LEFT TABLE EXT.	50%
*1		<b>1032 desk set 1</b>	<b>1x</b>
		length	EI
		width	EI
		height	EI
		material	EI
		color	EI
		quantity	EI
		pos. of table ext.	EI
	1	sp1 FIRST RIGHT/LEFT	
		quantity	2
		pos. of table ext.	right
	*1	sp2 FIRST RIGHT	100%
		pos. of table ext.	left
	1	sp3 FIRST REST	
		quantity	1



			quantity	2	
			quantity	3	
			pos. of table ext.	none	
			pos. of table ext.	right	
			pos. of table ext.	left	
	1	sp4	FIRST REST2		
			quantity	4	
			pos. of table ext.	none	
			pos. of table ext.	left	
<b>**1</b>		<b>1031</b>	<b>Desk set</b>		<b>1x</b>
<b>*2</b>		<b>1033</b>	<b>Desk set 2</b>		<b>1x</b>
			length	EI	
			width	EI	
			height	EI	
			material	EI	
			color	EI	
			quantity	EI	
			pos. of table ext.	EI	
	1	sp1	SECOND RIGHT/LEFT		
			quantity	2	
			pos. of table ext.	right	
	*1	sp2	SECOND RIGHT		100%
			pos. of table ext.	right	
	1	sp3	SECOND REST		
			quantity	2	
			quantity	3	
			pos. of table ext.	none	
			pos. of table ext.	right	
			pos. of table ext.	left	
	1	sp4	SECOND REST2		
			quantity	4	
			pos. of table ext.	none	
			pos. of table ext.	left	
<b>**1</b>		<b>1031</b>	<b>Desk set</b>		<b>1x</b>
<b>*3</b>		<b>1034</b>	<b>Desk set 3</b>		<b>1x</b>
			length	EI	
			width	EI	
			height	EI	
			material	EI	
			color	EI	
			quantity	EI	
			pos. of table ext.	EI	

	1	sp1	THIRD DESK		
			quantity	3	
			quantity	4	
<b>**1</b>		<b>1031</b>	<b>Desk set</b>		<b>1x</b>
<b>*4</b>		<b>1035</b>	<b>Desk set 4</b>		<b>1x</b>
			length	EI	
			width	EI	
			height	EI	
			material	EI	
			color	EI	
			quantity	EI	
			pos. of table ext.	EI	
	1	sp1	FOURTH DESK		
			quantity	4	
<b>**1</b>		<b>1031</b>	<b>Desk set</b>		<b>1x</b>
<b>*5</b>		<b>1036</b>	<b>phantom connecting segment</b>		<b>1x</b>
			width	EI	
			height	EI	
			material	EI	
			color	EI	
			quantity	E	
			shape	I	
	1	sp1	TWO DESKS		
			quantity	2	
	*1	sp2	CIRCLE		33%
			shape	circ	
	*2	sp3	TRIANGLE		33%
			shape	tria	
	*3	sp4	SQUARE		33%
			shape	squa	
	1	sp5	THREE DESKS		
			quantity	3	
	*1	sp3	TRIANGLE		50%
	*2	sp4	SQUARE		50%
	1	sp6	FOUR DESKS		
			quantity	4	
	*1	sp4	SQUARE		100%
	1	sp7	DESK LIGHT GRAY		
			color	lgra	
	*1	sp8	CONN. LIGHT GRAY		50%
			color	lgra	
	*2	sp9	CONN. BLACK		50%

		color	blac	
1	sp10	DESK GRAY		
		color	gray	
*1	sp11	CONN. GRAY		25%
		color	gray	
*2	sp9	CONN. BLACK		25%
*3	sp12	CONN. RED		25%
		color	red	
*4	sp13	CONN. BLUE		25%
		color	blue	
1	sp14	DESK BLACK		
		color	blac	
*1	sp8	CONN. LIGHT GRAY		20%
*2	sp11	CONN. GRAY		20%
*3	sp9	CONN. BLACK		20%
*4	sp12	CONN. RED		20%
*5	sp13	CONN. BLUE		20%
1	sp15	DESK BROWN		
		color	brow	
*1	sp16	CONN. BROWN		100%
		color	brow	
<b>**1</b>	<b>1026</b>	<b>connecting segment</b>		<b>1x</b>
<b>1</b>	<b>1031</b>	<b>Desk set</b>		
		length	EI	
		width	EI	
		height	EI	
		material	EI	
		color	EI	
		quantity	E	
		pos. of table ext.	EI	
		pos. of drawers	I	
		P-configuration	I	
1	sp1	NO TABLE EXT.		
		pos. of table ext.	none	
*1	sp2	DRAWERS LEFT		33%
		pos. of drawers	left	
*2	sp3	DRAWERS RIGHT		33%
		pos. of drawers	right	
*3	sp4	DRAWERS LE+RI		33%
		pos. of drawers	right	
1	sp5	TABLE EXT. LEFT		

		pos. of table ext.	left	
	*1	sp3	DRAWERS RIGHT	100%
	1	sp5	TABLE EXT. RIGHT	
	*1	sp2	DRAWERS LEFT	100%
	1	sp6	LENGTH 2m QUANTITY 1	
			length	200
			quantity	1
	*1	sp7	WITH P-Configuration	100%
			P-configuration	with
	1	sp8	QUANTITY 2,3,4	
			length	200
			length	180
			quantity	2
			quantity	3
			quantity	4
	*1	sp9	WITHOUT P-Configuration	100%
			P-configuration	w/o
	1	sp10	LENGTH 1.8m QUANTITY 1	
			length	180
			quantity	1
	*1	sp9	WITHOUT P-Configuration	100%
<b>*1</b>		<b>1017</b>	<b>desk</b>	<b>1x</b>
			length	EI
			width	EI
			height	EI
			material	EI
			color	EI
<b>**1</b>		<b>1014</b>	<b>side</b>	<b>2x</b>
			width	EI
			material	EI
			color	EI
<b>**2</b>		<b>1015</b>	<b>top</b>	<b>1x</b>
			length	EI
			width	EI
			material	EI
			color	EI
<b>**3</b>		<b>1016</b>	<b>connecting segment</b>	<b>1x</b>
			length	EI
			material	EI
			color	EI
<b>*2</b>		<b>1013</b>	<b>drawers</b>	<b>1x</b>
<b>*3</b>		<b>1030</b>	<b>phantom drawers</b>	<b>1x</b>

			material	EI	
			color	EI	
			pos. of drawers	E	
	1	sp1	LEFT+RIGHT		
			pos. of drawers	ri	
<b>**1</b>		<b>1013</b>	<b>drawers</b>		<b>1x</b>
<b>*4</b>		<b>1022</b>	<b>table ext. set</b>		<b>1x</b>
			material	EI	
			color	EI	
			pos. of table ext.	EI	
			with/without drawers	I	
	1	sp1	WITH TABLE EXT.		
			pos. of table ext.	left	
			pos. of table ext.	right	
	*1	sp2	WITH DRAWERS		50%
			with/without drawers	with	
	*2	sp3	WITHOUT DRAWERS		50%
			with/without drawers	w/o	
<b>**1</b>		<b>1021</b>	<b>table extension</b>		<b>1x</b>
			material	EI	
			color	EI	
			pos. of table ext.	EI	
<b>***1</b>		<b>1020</b>	<b>back</b>		<b>1x</b>
			material	EI	
			color	EI	
<b>***2</b>		<b>1019</b>	<b>side</b>		<b>2x</b>
			material	EI	
			color	EI	
<b>***3</b>		<b>1018</b>	<b>top</b>		<b>1x</b>
			material	EI	
			color	EI	
<b>**2</b>		<b>1013</b>	<b>drawers</b>		<b>1x</b>
<b>**3</b>		<b>1023</b>	<b>phantom drawers</b>		<b>1x</b>
	1	sp1	WITH DRAWERS		
			with/without drawers	with	
<b>***1</b>		<b>1013</b>	<b>drawers</b>		<b>1x</b>
<b>*5</b>		<b>1029</b>	<b>phantom connecting segment</b>		<b>1x</b>
			width	EI	
			height	EI	
			material	EI	
			color	EI	
			P-configuration	E	

			shape	I	
	1	sp1	WITH P-Configuration P-configuration	with	
	*1	sp2	CIRCLE		100%
<b>**1</b>		1026	connecting segment	circ	1x
<b>1</b>		1013	<b>drawers</b>		
			material	EI	
			color	EI	
			version	I	
	1	sp1	UNIVERSAL		
	*1	sp2	WITH NORMAL DRAWERS		50%
			version	norm	
	*2	sp3	WITH HANGING DRAWERS		50%
			version	hang	
<b>*1</b>		1007	<b>hanging drawer</b>		1x
			material	EI	
			color	EI	
			version	EI	
<b>*2</b>		1006	<b>pen drawer</b>		1x
			material	EI	
			color	EI	
<b>*3</b>		1002	<b>frame</b>		1x
			material	EI	
			color	EI	
<b>**1</b>		1001	<b>side</b>		2x
			material	EI	
			color	EI	
<b>**2</b>		1003	<b>bottom</b>		1x
			material	EI	
			color	EI	
<b>**3</b>		1004	<b>back</b>		1x
			material	EI	
			color	EI	
<b>**4</b>		1005	<b>front</b>		1x
			material	EI	
			color	EI	
<b>*4</b>		1010	<b>drawer</b>		1x
			material	EI	
			color	EI	
<b>*5</b>		1011	<b>rail</b>		1x

			version	EI	
	1	sp1	NORMAL DRAWERS		
			version	norm	
<b>*6</b>		<b>1009</b>	<b>lock</b>		<b>1x</b>
<b>1</b>		<b>1026</b>	<b>connecting segment</b>		
			width	EI	
			height	EI	
			material	EI	
			color	EI	
			shape	EI	
<b>*1</b>		<b>1024</b>	<b>side</b>		<b>2x</b>
			width	EI	
			height	EI	
			material	EI	
			color	EI	
<b>*2</b>		<b>1025</b>	<b>top</b>		<b>1x</b>
			width	EI	
			material	EI	
			color	EI	
			shape	EI	

## References

---

- AXSA86            Axsater,S.,*On the feasibility of aggregate production plans*  
 , Operations Research ,34-5,796-800, 1986
- BARK89            Barker,V.E.,D.E. O'Conner,*Expert systems for configuration*  
*at Digital: XCON and beyond* , Communication of the ACM  
,32-3,298-318, 1989
- BEME86            Bemelmans,T.M.A.,*Bedrijfskundig ontwerpen van bestuurlijke*  
*informatiesystemen* , Automatisering met een menselijk  
gezicht,Kluwer, 1986
- BERT90            Bertrand,J.W.M.,J.C. Wortmann,J. Wijngaard,*Production*  
*control: a structural and design approach*,Elsevier Science  
Publishers,322, 1990
- BERT92            Bertrand,J.W.M.,J.C. Wortmann,J. Wijngaard,*A framework for*  
*production control and materials management* , Integration in  
Production Management Systems,IFIP-Transactions,213-222,  
1992
- BURB90            Burbidge,J.L.,*Production control: a universal conceptual*  
*framework* , Production planning & control ,1-1, 1990
- CARR76            Carruthers,J.M.,*Product Structure Charts: Key to designing*  
*and controlling modular bills-of-material* , APICS Conference  
Proceedings, 1976
- CODA82            Beitz,E.H.,N.H. Buck,P.C. Jorgensen,*A modern appraisal of*  
*decision tables : a codasyl report*,ACM,300, 1982
- DANI                Daniels,G.,*Een geïntegreerd industrieel productieplanning- en*  
*besturingssysteem met een generieke produktstructuur* ,  
IOP-Bouw Seminar and Lecture,16, 0



- 
- DONS87            Donselaar,K. van,F. Jenniskens,J. Timmer,*Line requirements planning alternatief voor MRP ? (I)* , Tijdschrift voor Inkoop & Logistiek ,3-11,21-25, 1987
- DONS89            Donselaar,K. van,*Material Coordination under uncertainty* , Dissertation,Eindhoven Univ. of Technology, 1989
- DOOR92            Dooren,E.P.H.M.,*Een aanzet tot een verbeterde informatie-voorziening*, 1992
- DUMO88            Dumoulin,W.J.E.,*Generieke stuklijsten bij de assemblage van TV-toestellen*, 1988
- EEKH93            Eekhout,M.M.I.M.,*Materiaalplanning met generieke stuklijsten : een prototype* , afstudeerverslag,TUE/BDK/I&T, 1993
- GERA89            Geraerds, W. M. J, M. Igel,*Flexibiliteit in logistiek*,Samson/Nive,356, 1989
- HIGG92            Higgs,P.,J. Browne,*Master production scheduling: a concurrent planning approach* , Production Planning & Control ,3-1,2-18, 1992
- HOEK91            Hoekstra,Sjoerd,Jac Romme,*Integral logistic structures*,McGraw-Hill,164, 1991
- JONG91            Jong,H.A.L.G.de,*Genereren van ordergebonden views op de stuklijst*, 1991
- KNOX84            Knox,Charles S.,*Engineering documentation for CAD/CAM applications*,Dekker New York,307, 1984
- KORN92            Kornelius,L.,C. Kreuwels,P. van der Vlist,*Aspects of the External Integration of Production Management Systems with*

- 
- EDI* , Integration in production management systems,Elsevier Science Publishers,175-182, 1992
- KREU94            Kreuwels,C.M.A.,*Externe logistieke integratie en EDI*,Kluwer Techniek,237, 1994
- LAND89            Landvater,Darryl V.,Chr. D. Gray,*MRP II standard system : a handbook for manufacturing software survival*,Essex Junction : Oliver Wight,315, 1989
- LANS92            Lansink,H.A.R.M.,*Generieke stuklijsten bij vlisco B.V.?*, 1992
- LEEN91            Leentjes,F.,*Onderzoek naar de toepasbaarheid van een generiek stuklijstconcept bij SKF transportwielen*, 1991
- LEON87            Leonard-Barton,D.,*The case for integrative innovation: an expert system at Digital* , Sloan Management Review ,Fall,7-19, 1987
- MATH82            Mather,H.,*Bills of material recipes and formulations*,Wright Publishing Company Inc., 1982
- MOUL89            Dumoulin,W.J.E.,*Case bureau-eilanden* , Interne notitie,TUE/BDK/BISA, 1989
- ORLI72            Orlicky,J.A.,G.W. Plossl,O.W. Wight,*Structuring the Bill of Material for MRP* , Production and Inventory Management ,13-4,9-17, 1972
- RIJN85            Rijn,Th.M.J.van,*Een typologie van productie-eenheden als gereedschap voor diagnose* , Informatie ,27,363-374, 1985
- RIJN86            Rijn,Th.M.J.,*Producersen door informeren Informatie-eisen voor verschillende produktiesituaties*,Kluwer, 1986

- 
- ROOD92 Rooden,M.A.H.,*Herontwerp van de logistieke organisatie in het voortraject van Sch lumberger Bladel*, 1992
- SARI81 Sari,F.J.,*The MPS and the Bill of Material go hand-in-hand*,Richard C. Ling Inc.,36, 1981
- SCHO85 Schonsleben,P.,*Flexibele Produktionsplanung und -Steuerung mit dem Computer*,CW-Publikationen, Munchen, 1985
- SCHR90 Schroder,E.J.B.M.,*Generieke stuklijsten casus*, 1990
- SLOB93 Slob,E.J.,*Generieke produktstructuren bij Philips Medical Systems* , afstudeerverslag,TU Eindhoven,65, 1993
- VEEN91 Veen,E.A. van,*Modelling product structures by generic bills-of-material* , Dissertation,Tech. Universteit Eindhoven, 1991
- VEN91 van de Ven,H.P.,*Generieke stuklijst t.b.v. productie*, 1991
- VOLL88 Vollmann,T.E.,W.L. Berry,D.C. Whybark,*Manufacturing planning and control systems*,Dow Jones-Irwin, Illinois, 1988
- WEDE81 Wedekind,H.,T. Muller,*Stucklistenorganisation bei einer grossen Variantenzahl* , Angewandte Informatik ,9,377-383, 1981
- WEMM84 Wemmerlov,U.,*Assemble-To-Order manufacturing: implications for materials management* , J. of Operations Management ,4-4,347-368, 1984
- WIJN87 Wijngaard,J.,*Production control in a consumer electronics factory* , Engineering Costs and Production Economics ,12,165-173, 1987

- YNTE92            Yntema,E.,*Family structures methoden om het produktvernieuwingsproces te ondersteunen*, 1992

## Notational conventions for data structure diagrams

An entity type is represented by a rectangle (see Figure 1). Each line represents a relationship between two entity types. The cardinality is indicated by a special symbol at the point where the line is connected to entity type B. These symbols are explained in Figure 1 where the minimum and maximum numbers of entities of type B related to the type A entity are shown for each symbol.

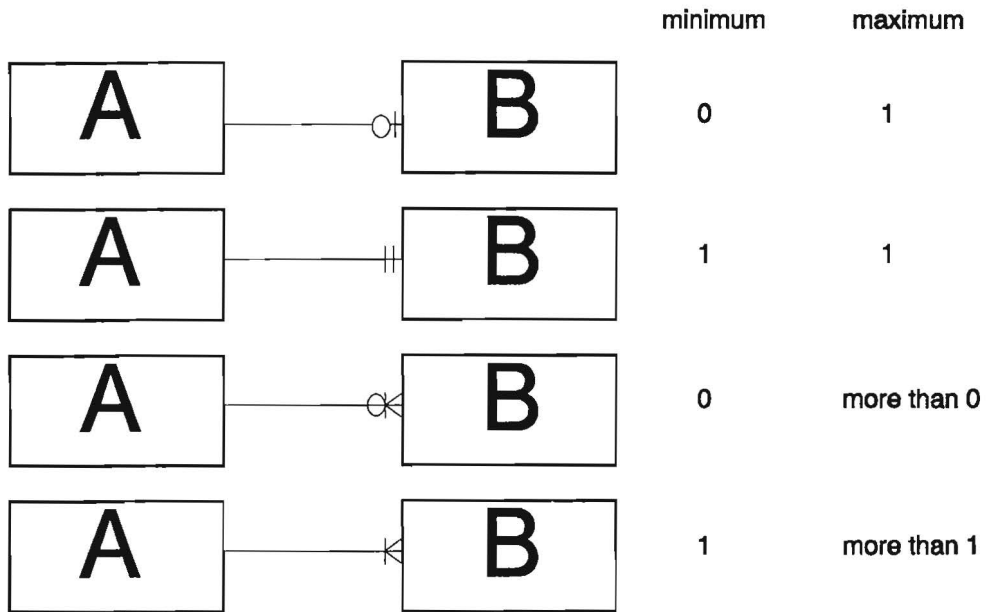


Figure 1: Cardinality symbols

*bill-of-material generator*: a system that produces a bill-of-material of a variant based upon the identification

*configurator*: a system that supports the identification of a variant

*direct identification*: an identification via the product code

*external parameter*: a parameter of a top level specialty

*generic bill-of-material (GBOM)*: a bill-of-material of a whole product family

*generic primary product (GPP)*: a generic product at the lowest level in the generic bill-of-material

*generic product (GP)*: a set of products (a product family)

*generic material requirements planning (GMRP)*: a system for material requirements planning based upon the generic bill-of-material

*generic subassembly (GS)*: a generic product at an intermediate level in the generic bill-of-material

*identification*: the determination of a product's identity

*indirect identification*: an identification through the use of a bill-of-material or a partial bill-of-material

*inheritance*: the transfer of a chosen parameter value from a common parent to the relevant generic primary product's

*internal parameter*: a parameter of a lowest level specialty

*lowest level specialty*: a specialty at the lowest level of a decision tree

*multivariant*: a set with multiple variants

*own parameter*: a parameter of generic product of which the parameter value must be chosen at the generic product level

*parameter subrange*: a range of possible parameter values

*primary product*: a product at the lowest level of a bill-of-material

*product set*: a generic product or a specialty

*pseudo product*: a set of products that is included in a modular bill-of-material

*specialty*: a subset of a generic product usually by specifying the values of one or more parameters of that generic product

*top level specialty*: a specialty at the top level of a decision tree

*universal top level specialty*: a top level specialty including all of the valid variants of a generic product

*variant*: a product within a generic product

*version of a product*: a modified product that is interchangeable with the original product

The primary focus of the research presented here is the improvement of information systems used to support production control within industrial companies. In connection with this, this study is oriented toward companies that produce and supply complex products with many possible variants.

This thesis is divided into four parts. Part I addresses the general problem area of production control in companies that deal with many product variants. A requirements specification for a product model is developed at the end of Part I. These requirements also include the two most important applications within the product model to be developed, namely, the configurator and the material requirements planning applications. In Part II, a product model in the form of a generic bill-of-material is developed for product families. The configurator and the material requirements planning (GMRP) are described in Part III. Finally, in Part IV, an evaluation of the three different approaches is provided.

Part I covers Chapters 1, 2 and 3 of this thesis. The environment within which the product model is to be applied is defined in Chapter 1. The production typology proposed by Burbidge is used as the basis for this definition. The production environment identified for the product model to be developed can be characterized as one in which either a batch-oriented or continuous production process takes place. In addition, the target environment is seen as one in which the finished products are made from a large number of purchased components and, thus, need to be "exploded." The method of production control can be characterized based upon the location of the customer order decoupling point (CODP) in the flow of materials. The environment that is most suited to this study has its CODP at the finished products inventory (Make-To-Stock) or at the inventory point preceding the assembly stage (Assemble-To-Order). The conclusion drawn at the end of this chapter is that the most important modules of an information system to support this type of environment are the module to produce product documentation, the configurator to identify and specify a product variant and the material requirements planning module.

Chapter 2 discusses the issues that arise when the variety of finished products is expanded. The consequences of moving from a pure MTS environment to a mixed ATO/MTS environment are analyzed with respect to the primary process, the control aspects and the information system. The most important conclusion is that



the information system in this type of environment should be able to work with aggregated information with respect to the master production scheduling, material coordination and work order issuing functions.

In Chapter 3, the objective of this research is first formulated as follows:

*The objective of the research is to improve the information systems that support production control in industrial environments. The research focuses particularly on the problems encountered by companies that produce and supply complex products with many possible variants. This type of production situation is often characterized as being an assemble-to-order/make-to-stock environment.*

Subsequently, based upon this objective, the design problem to be addressed is defined in the following way:

*Design a product model for a product family for use in an assemble-to-order/make-to-stock production environment. Using this model, it should be possible to configure a specific product so that it can be manufactured and the material requirements can be planned appropriately.*

This means that three separate design issues need to be resolved, namely: designing a product model (generic bill-of-material), designing a configurator and planning the material requirements. A requirements specification is then developed for these design problems. A test case developed specifically for this study is included in this chapter.

Part II consists of Chapters 4, 5, 6 and 7. A number of basic terms related to the bill-of-material, the traditional product model, are explained in Chapter 4. The concept of a generic bill-of-material is developed in Chapter 5. The foundation of this concept is the fact that a finished product variant in an ATO/MTS environment is determined by the component variants at the lowest level in the product structure. A generic product is a set of products, also referred to as variants. In order to identify and specify a variant within such a set, three approaches are described, namely: directly without parameters, directly with parameters and indirectly with parameters. A restriction is imposed in this chapter with respect to choosing parameter values whereby the parameter values for the components of

---

products must be chosen in the same way. The requirement here is that the parameter values of a parent must be transferable to a component. This is referred to as the inheritance mechanism. The term specialty is also introduced in this chapter. A specialty is a subset of a generic product that is characterized by one or more parameter values.

A number of methods for keeping track of requirements and conditions within a generic bill-of-material are described at the beginning of Chapter 6. Then the possibility of using continuous parameters in addition to discrete parameters is introduced. The possible use of algorithms as part of the decision process is proposed in order to use these continuous parameters more effectively and to extend the applicability of a generic bill-of-material. Finally, an approach to documenting engineering changes within a generic bill-of-material is discussed.

The structure of a generic bill-of-material and using a generic bill-of-material are described in more detail in Chapter 7.

Part III consists of Chapters 8 and 9 where two designs for applications based upon a generic bill-of-material are presented. The configurator application is described in Chapter 8. A configurator is a system designed to support the identification and specification of a product variant.

The generic MRP (GMRP) application is described in Chapter 9. GMRP uses algorithms from Line Requirements Planning (LRP) and Material Requirements Planning (MRP). GMRP focuses on aggregate planning at the finished product level as well as at the component level. When the flow of materials needs to be initiated, the focus is then shifted to the specific variants to be produced. This means that a number of factors need to be taken into account and recorded at the aggregate level as well as at the variant level. In a mixed ATO/MTS environment, the fast-movers within a product family can be planned at the variant level and the slow-movers planned at the aggregate level.

The study in its entirety is evaluated in Part IV. The generic bill-of-material is compared with a number of alternative product models in Chapter 10.

Het hoofdonderwerp van dit onderzoek is het verbeteren van informatiesystemen ter ondersteuning van de productiebesturing voor industriële ondernemingen. Daarbij richt zich het onderzoek op ondernemingen met complexe producten, die in veel varianten worden geleverd.

Dit boek bestaat uit een viertal delen. In deel I wordt de problematiek van de productiebesturing uitgewerkt ten aanzien van ondernemingen met veel varianten. Op het einde van deel I wordt een eisenblad opgesteld voor een productmodel en voor de twee belangrijkste toepassingen van het te ontwerpen productmodel nl de configurator en de materiaalbehoefteplanning. In deel II wordt een productmodel ontwikkeld voor productfamilies, de generieke stuklijst. In deel III worden de configurator en de materiaalbehoefteplanning (GMRP) beschreven. In deel IV tenslotte wordt een evaluatie gegeven van de drie ontwerpen.

Deel I bestaat uit de hoofdstukken 1,2 en 3. In hoofdstuk 1 is een afbakening gemaakt van de omgeving, waarin het productmodel kan worden toegepast. Bij deze afbakening is er gebruik gemaakt van de productietypologie van Burbidge. De voor het te ontwikkelen productmodel relevante productieomgeving heeft als karakteristieken, dat het seriematig of continue produceert. En tevens, dat het explosief. Dat wil zeggen, dat men eindproducten produceert uit vele inkoopdelen. De productiebeheersing is gekarakteriseerd met behulp van de plaats van klantorderontkoppelpunt (KOO) in de goederenstroom. Bij de voor dit onderzoek relevante omgeving ligt het KOO bij het voorraadpunt eindproduct (Make-To-Stock) of bij het voorraadpunt vlak voor de assemblage (Assemble-To-Order). Op het einde van dit hoofdstuk wordt geconcludeerd, dat de belangrijkste modules van een informatiesysteem voor deze omgeving zijn de productbeschrijvingsmodule, de configurator ter identificatie van een variant en de materiaalbehoefteplanningsmodule.

In hoofdstuk 2 gaat met name in op de problematiek, die ontstaat bij het vergroten van de variëteit aan eindproducten. Met name zijn de gevolgen voor de overgang van een pure MTS-omgeving naar een gemengde ATO/MTS omgeving onderzocht voor het primaire proces, voor de besturing en het informatiesysteem. De belangrijkste conclusie is, dat het informatiesysteem in dergelijke omgeving moet kunnen werken met geaggregeerde informatie bij met name de functies hoofdproductieplan, materiaalcoördinatie en werkorderuitgifte.

---

In hoofdstuk 3 wordt allereerst het doel van dit onderzoek geformuleerd. Dit doel is als volgt omschreven:

*Het doel van het onderzoek is het verbeteren van de informatiesystemen ter ondersteuning van de productiebesturing voor industriële ondernemingen. Met name richt zich het onderzoek op de ondernemingen met complexe producten, die in veel varianten worden geleverd. Daarbij kan de productieomgeving worden getypeerd als een assemble-to-order/make-to-stock omgeving.*

Vervolgens is op basis van deze doelstelling een ontwerpvraag opgesteld. Deze ontwerpvraag luidt:

*Ontwerp een productmodel voor een productfamilie bestemd voor een assemble-to-order/make-to-stock productieomgeving, waarmee het mogelijk is om een specifiek product te kunnen configureren, om een specifiek product te kunnen produceren en om de materiaalbehoeften daarmee behoorlijk te kunnen plannen.*

Dit betekent, dat er een drietal ontwerpen moeten worden gemaakt nl. een productmodel (generieke stuklijst), een configurator en een materiaalbehoefteplanning. Voor deze ontwerpen zijn vervolgens een eisenblad gegeven. In de appendix van dit hoofdstuk is een voor dit onderzoek ontwikkelde testcase gegeven.

Deel II bestaat uit de hoofdstukken 4,5,6 en 7. In hoofdstuk 4 worden een aantal basisbegrippen van de stuklijst, het traditionele productmodel, gegeven. In hoofdstuk 5 wordt het generieke stuklijstconcept ontwikkeld. Het basisidee van de generieke stuklijst is het feit, dat een eindproductvariant in een ATO/MTS omgeving wordt bepaald door de componentvarianten op het laagste nivo van de productopbouw. Een generiek product is een verzameling van producten, ook wel varianten genoemd. Om een variant binnen een dergelijke verzameling te identificeren worden er drie mogelijkheden geïntroduceerd nl. de directe zonder parameters, de directe met parameters en de indirecte identificatie met parameters. In dit hoofdstuk wordt een type beperking in de keuzes van parameterwaarden gegeven nl. dezelfde keuze van parameterwaarden bij componenten van product. Hierbij is het noodzakelijk om parameterwaarden van een parent te kunnen doorgeven aan een component. Dit wordt het overervingsmechanisme genoemd. In

dit hoofdstuk wordt ook het begrip specialiteit geïntroduceerd. Een specialiteit is een deelverzameling van een generiek product, die gekenmerkt wordt door een of meer parameterwaarden.

In hoofdstuk 6 zijn allereerst een aantal methodes behandeld om voorwaarden in de generieke stuklijst vast te leggen. Vervolgens is de mogelijkheid geïntroduceerd om continue parameters naast discrete parameters te kunnen gebruiken. Om effectiever gebruik te kunnen maken van deze continue parameters en toepassingsgebied van de generieke stuklijst te verbreden is het mogelijk gebruik te maken van algoritmes in het keuzeproces geïntroduceerd. Tenslotte is het vastleggen van technische wijzigingen bij de generieke stuklijst uiteengezet.

In hoofdstuk 7 is er beschreven, hoe een generieke stuklijst moet worden opgebouwd en waar een generieke stuklijst kan worden toegepast.

Deel III bestaat uit de hoofdstukken 8 en 9, waarin een tweetal toepassingen van de generieke stuklijst worden behandeld. In hoofdstuk 8 wordt de configurator beschreven. Een configurator is een systeem, dat ondersteuning biedt voor het opstellen van de identificatie van een variant.

In hoofdstuk 9 wordt de generieke MRP (GMRP) beschreven. GMRP maakt gebruik van algoritmes afkomstig van Line Requirements Planning (LRP) en Material Requirements Planning (MRP). GMRP blijft zolang mogelijk in aggregaten plannen, zowel op eindproducten- als op componentenniveau. Als de goederenstroom daadwerkelijk op gang moet worden gebracht, zal men moeten overgaan op de varianten zelf. Dat betekent, dat een aantal grootheden zowel op aggregaat- als op variantniveau moet worden vastgelegd. Bij gemengde ATO/MTS omgeving kan men de snellopers van een productfamilie per variant plannen en de langzaamlopers per aggregaat plannen.

Deel IV wordt het geheel van het onderzoek geëvalueerd. In hoofdstuk 10 wordt de generieke stuklijst vergeleken met een aantal productmodellen.

## Index

bill-of-material	36	generic bill-of-material	51
exploded list	37	Van Veen	178
generator	119	generic MRP	139
generic	51	generic primary product	52
imploded list	37	generic product	51
manufacturing	144	generic subassembly	53
modular	142	GPP coordination	56
parts list	35	identification	47
bill-of-material relationship	35	complete tree	132
component	35	direct	36, 47
end point	35	incomplete	58
parent	35	indirect	36, 47
quantity per	35	product code	34
sequence number	35	imploded list	37
starting point	35	indirect identification	36, 47
component	35	information system	1
configurator	119	inheritance	56
cover calculation	193	integral planning	154
data registration	1	interchangeability	42
decision table	90	internal parameter	64
complete	92	level	42
consistency	92	Line Requirements Planning	154
decision rule	90	logical graph	173
limited input	91	lowest level code	42
direct identification	36, 47	lowest level specialty	64
end point BOM	35	LRP	
engineering change		line requirements planning	154
interchangeability	42	manufacturing BOM	144
version	43	Master Production Schedule	141
exploded list	37	modular BOM	142
external parameter	61	multivariant	151
fast-mover	167	option	13, 142
feature	13, 142	overplanning GMRP	164
Final Assembly Schedule	141	own parameter	63
finished product	33	parameter	49
forecast consumption	156	continue	129
generative systems	15	external	61

---

internal	64	premise	94
own	63	pseudo product	142
subrange	95	quantity per	35
translation	80	quotation time fence	159
parent	35	rule base	181
parts list	35	run-out time	163
path	41	sequence number	35
path-length	42	slow-mover	167
planning module	13, 142	specialty	57
planning percentage	142	lowest level	64
primary product	34	top level	61
product	34	universal top level	65
fast-mover	167	starting point BOM	35
finished	33	subassembly	34
generic	51	top level specialty	61
generic primary	52	translation parameters	80
generic subassembly	53	universal top level specialty	65
primary	34	variant	47
pseudo	142	version	43
slow-mover	167	version number	43
subassembly	34		
variant	47		
product code	34		
product family bill-of-material	49		
product family	47		
product family bill-of-material set	49		
product family bill-of-material system	49		
product set	149		
multivariant	151		
product structure	33		
product structure chart	172		
production environment			
assemble-to-order	140		
make-to-stock	140		
production rule	94, 181		
action	94		

## Curriculum vitae

---

De auteur van dit proefschrift werd op 26 april 1941 geboren te 's-Hertogenbosch. Hij is afgestudeerd als bedrijfseconometrist aan de Katholieke Universiteit Brabant, en was daarna werkzaam bij Océ van der Grinten N.V. te Venlo. Bij dit bedrijf heeft hij drie functies vervuld.

Zijn eerste functie was systemconsultant bij de E.D.P. afdeling. Hij gaf leiding aan automatiseringsprojecten op het gebied van productiebesturing en besturing van physical distribution. Een van deze projecten was het invoeren van een MRP-I besturing voor een assemblage fabriek van kopieerapparaten.

Vervolgens werd hij in 1975 benoemd tot chef New Systems (14 systeemanalisten /programmeurs) bij de E.D.P. afdeling en was verantwoordelijk voor alle nieuw te bouwen systemen.

In 1976 aanvaardde hij een functie bij de afdeling Corporate Strategy en Group Controlling, een stafafdeling van de Raad van Bestuur. Hij ontwikkelde en beheerde daar ondermeer een bedrijfsmodel voor de gehele onderneming en was tevens nauw betrokken bij het opzetten van een planningscyclus binnen de onderneming.

In 1980 trad hij in dienst van het Instituut voor Hoger Beroeps Onderwijs te Venlo als adjunct directeur om leiding te geven aan de startende Vervoersacademie (een logistieke HTS opleiding).

Sinds 1987 is hij in dienst van de faculteit Technische Bedrijfskunde aan de Technische Universiteit Eindhoven, in de vakgroep Information & Technology.



# STELLINGEN

behorende bij het proefschrift

INTELLIGENT

PRODUCT FAMILY DESCRIPTIONS

FOR

BUSINESS APPLICATIONS

HERMAN M.H. HEGGE

## I

Bij de beschrijving van een product moet een duidelijk onderscheid gemaakt worden tussen de identificatie en de specificatie van een product (hoofdstuk 5 van dit proefschrift).

## II

Een criterium om de modulariteit van een product familie ontwerp te beoordelen is de componeerbaarheid. Stel, dat  $N_i$  het aantal varianten is van de generieke component  $i$  op het laagste nivo in de productopbouw en  $k$  het aantal generieke componenten op dat laagste nivo. Dan is een maat voor componeerbaarheid in een ATO/MTS omgeving:

aantal maakbare varianten van de productfamilie

---

$$N_1 \times N_2 \times \dots \times N_k$$

## III

De stuklijst is een variant van de generieke stuklijst d.w.z. de functionaliteit van de stuklijst is een deel van de functionaliteit van de generieke stuklijst (hoofdstuk 6 van dit proefschrift).

## IV

MRP is een variant van GMRP (hoofdstuk 9 van dit proefschrift).

## V

LRP is een variant van GMRP (hoofdstuk 9 van dit proefschrift).

## VI

De levensduur van een product architectuur dient langer te zijn dan van een productfamilie.

## VII

Bij het ontwerpen van een productfamilie moet het hergebruik van componentfamilies voorop staan in plaats van het hergebruik van componenten.

## VIII

Ketenlogistiek is een pleonasme.

## IX

De huidige standaard software pakketten bezitten een grote functionaliteit. Daardoor wordt de gebruiker geconfronteerd met overbodige uitgaven.

## X

Een standaard software pakket met veel functionaliteit, die door weinig gebruikers wordt benut, dient als een productfamilie te worden ontworpen.

## XI

De mate van tevredenheid waarin de mens verkeert staat vaak in schril contrast met zijn pakket aan mogelijkheden.

## XII

Voor vele mensen zijn vergelegen oorden nabij en nabijgelegen oorden veraf.