Generative bill of material processing systems

PRODUCTION PLANNING & CONTROL, 1992, VOL 3, NO 3, 314–326

Generative bill of material processing systems

E. A. VAN VEEN and J. C. WORTMANN

Abstract. Many manufacturing companies have, over the past several years, been forced to drastically increase their product assortment Many of them have found that traditional bill of material (BOM) processing systems do not sufficiently support the maintenance of very large amounts of product (structure) data. This maintenance problem is implied by the requirements to identify each product variant explicitly by a part number for which product data including BOM data must be defined explicitly. With the introduction of MRP oriented production control systems a great deal of literature has been devoted to structuring the BOMs to support the master production scheduling function. In particular the concept of modularization BOMs is a well known research issue. In this paper the assumptions underlying the successful modularization of BOMs are made explicit. It will be shown that in practice, seldom can all assumptions be true. Therefore, new concepts for representing BOMs of large numbers of product variants are required as well as the supporting BOM-processing systems. A new kind of BOM-processing systems is introduced, called generative BOM-processing systems. One particular kind of generative BOM-processor is discussed in detail, i.e. the variant BOM-processor. Although the concept underlying the variant BOM-processor does solve some of the problems of maintaining large numbers of final product variants and their BOMs, it has a number of limitations An analysis of these limitations shows that they are implied by the strong focus on maintaining *final* product variants

1. Introduction

In recent years, a substantial increase in product variety and customization as well as a shortening of product life cycles has been showing itself both in markets for capital goods and in markets for consumer goods. We will use the term *product flexibility* for these phenomena. Product flexibility is becoming a new factor in competition, which has an enormous impact on product design, production facilities, control systems and supporting information systems. Traditional production control information systems (PCIS) are burdened by the growing number of products which need to be controlled and the growing amount of related product data which must be handled. Many problems arise in defining and maintaining product data. These problems affect the performance of PCIS with regard to aspects such as data

Authors: E. A. van Veen, DAF Trucks, Postbus 90065, NL-5600 PT, Eindhoven, The Netherlands; Johan C. Wortmann, Eindhoven University of Technology, PO Box 513, NL-5600 MB Eindhoven, The Netherlands.

E. A. VAN VEEN, graduated in Industrial Engineering and Management Science in 1986 During the period 1986–1990 he carried out a doctoral study on modelling product structures, at the Eindhoven University of Technology, while being detached from the TNO Institute for Production and Logistics Research. Since 1990 he has been the Product Development Information Manager of the van and truck manufacturer DAF in Eindhoven, The Netherlands

JOHAN C. WORTMANN studied industrial engineering and management science at Eindhoven University of Technology He has been active in the development of information systems for production/inventory control since 1973 and wrote a doctoral thesis on the subject. He has been involved in a number of practical applications, both in component manufacturing and in assembly operations. He worked in the first half of 1985 as visiting professor at Rutgers University, New Jersey on databases. Dr Wortmann is currently employed at Eindhoven University of Technology as professor of information systems for production management.

entry support, retrieval functions and the correctness and timeliness of provided information As a consequence the overall logistic performance may deteriorate

Most PCIS rely heavily on the descriptions of products, production facilities and production processes The majority of the currently available standard software for production control is based on the manufacturing resources planning (MRP II) concept (Vollmann *et al.* 1988) In this paper we:

- describe traditional concepts for recording product data and product structure data in PCIS, and analyse why they fail to perform satisfactorily if product variety increases strongly;
- introduce the generative bill of material concept which is specifically suited to deal with large product variety;
- discuss an existing generative bill of material concept called the variant bill of material concept.

In Section 2 of this paper, we describe the representation of BOMs in traditional PCIS In Section 3 we explain that, as product variety becomes wider, traditional concepts for representing product and product structure data become less suitable. Section 4 describes the technique of modularizing bills of material Section 5 introduces the general architecture of generative bill of material processing systems. In Section 6 an existing generative bill of material concept, the variant bill of material, is explained in detail. Section 7 deals with two principle problems associated with the variant bill of material concept Finally, Section 8 summarizes our conclusions.

In a subsequent paper, titled 'New developments in generative BOM-processing systems' (1992), we will introduce a new generative bill of material concept which deals with the problems associated with the variant BOM concept This concept is called the generic bill of material concept

2. Product data and product structure data

In a manufacturing business, many different products must be controlled. Not only final and purchased products are important, but also many semi-finished products such as sub-assemblies are important to recognize and control. Many products are related to each other in the way that the one product is required to manufacture another.

The way in which a product is built up from purchased parts and/or semi-finished products (which in turn consist of other semi-finished products and/or purchased

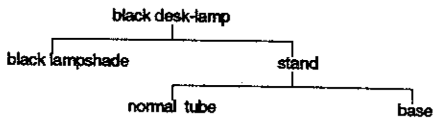


Figure 1 Product structure of a desk-lamp.

parts) is called the *product structure* of that product. Figure 1 depicts the product structure of a final product *black desk-lamp* The relationships between the products in a product structure are so-called *gozinto-relationships* They represent the fact that a product is consumed in the process of manufacturing or assembling another product. The product *C* which is *consumed* in the gozinto-relationship with product *P* is called the *component product* The product *P* which *consumes* product *C* is called the *parent product* in the gozinto-relationship with *C*.

The introduction of MRP and other material planning concepts based on the principle of co-ordinated material planning, made product structure data become more and more important to the material planning function. These planning concepts rely on gozinto-relationships for calculating planned material requirements for component products based on planned material requirements of parent products. Therefore most PCIS which are based on material co-ordination rely heavily on the representation of the product structure of the products to be planned.

The representation of the set of all gozinto-relationships of a parent product *P* is called the *bill of material (BOM) of P*. *Product data* are data on single products *Product structure data* are data on the gozinto-relationship between two products. In PCIS, these different types of data are commonly represented by means of two separate entity-types, namely:

- product
- bill of material relationship

Figure 2 depicts the typical data structure diagram of a BOM-processor used in most currently available PCIS (for conventions on data structure diagram notations see Appendix 1)

A product which is the parent product in a particular

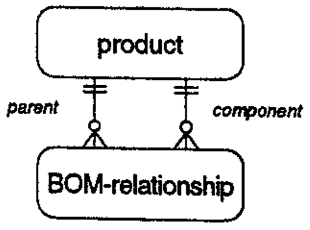


Figure 2 Typical data structure diagram of a BOM processor.

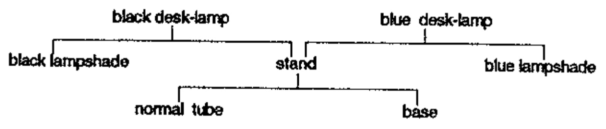


Figure 3 BOM structures of two desk-lamps

BOM-relationship may be the component product in one or more other BOM-relationships Thus, if a product S is considered as a parent product, it may not only consist of one or more component products S may also, if considered as a component product, be applied in one or more different parent products. Still the BOM of S is only recorded once in the product database, independent of the number of different parent products in which S is applied as a component product. This principle for recording BOMs in PCIS is commonly referred to as *modular storage of a BOM*. Notice carefully that the term modular refers here to the way in which product structure data is *stored*. In Section 4 the term modular will be used again but in a completely different context, namely with respect to the way in which BOMs are *structured*.

Figure 3 depicts the BOM-structures of two different final products *black desk-lamp* and *blue desk-lamp*.

Tables 1 and 2 show the entities of the type *product*, respectively entities of the type *BOM-relationship*, which establish these BOMs.

The product *stand* (567) is the parent product in the BOM-relationships with the products *normal tube* and *base*. However, in the BOM-relationships with the two desk-lamps, it plays the role of a component product The modular storage principle allows the BOM-relationships of the stand to be defined only once, although it is applied in two higher level products

Table 1. Product entities

Product number	Product description
123	Black desk-lamp
234	Blue desk-lamp
345	Black lampshade
456	Blue lampshade
567	Stand
678	Normal tube
789	Base

Table 2. BOM-relationship entities.

Parent product	Component product	Quanity/per
123	345	1
123	567	1
234	456	1
234	567	1
567	678	1
567	789	1

3. Recording product data and BOM data: a dilemma

A wider product assortment provides more product variety to the sales function However, it will also become more difficult to handle the increasing amount of information on the entire product assortment In many kinds of industries the number of different final products may easily exceed a million Sales catalogues which are based on enumerating unique part numbers will then no longer be feasible More complicated catalogue systems will be required to support the unique description of a product variant for the purpose of order entry and to support the selection of a product for a particular customer

Yet another problem area concerns the definition and maintenance of BOM data for the enormous number of different products. If the number of different products is too large to define and maintain a separate part number for each different product variant, then also no separate BOMs for these products can be defined and maintained We will explore this problem in greater detail in the subsequent sections. In Section 4, the typical MRP- solution for reducing the effort to create and maintain order-independent BOM data are analysed Furthermore, Appendix 2 describes a traditional method, called the add-and-delete technique, for reducing numbers of BOM-relationships to be stored

4. Modular bills of material

4 1 *The concept of modularization*

In traditional MRP literature the problem of large product variety is mainly approached from the viewpoint of forecasting and master production scheduling (see Orlicky *et al* 1972, Orlicky 1975, Mather 1982, Kneppelt, 1984, Vollmann *et al* 1988) However, it is often argued that modular BOMs are also a solution to the problem of maintaining extremely large numbers of different BOMs of final products In order to analyse the value of the technique of modularizing BOMs, the key concepts of this technique, namely option and planning module, need to be defined precisely

We define an *option* as the representation of a property of a product, such as blue, black or luxury The concept *feature* is introduced to represent a set of mutually exclusive options (e g the feature *colour* and the options *blue*, *red*, *black* and the feature *type* and the options *normal*, *flexible*) (Wortmann 1987) The features and options within a product family are defined in such a way that they can be applied to establish unique, identifying product descriptions A *planning module* defines a set of component products which are required to realize a final product with a particular (combination of) option(s), or to realize any product within a product family (in case the planning module defines the component products which are common to all final products of the product family) It is often assumed that a planning module equals an option Under this assumption, options (or planning modules) can be applied to uniquely identify a final product and at the same time to constitute the BOM of that final product

The modularization of traditional BOMs consists of creating planning modules given product families, features and options We will illustrate this technique with the example of the desk-lamps Assume a product family of six desk lamps In this family there are two different options for the stand, namely *normal* and *flexible* and three different colour options, namely *blue*, *black* and *white* The *type* and *colour* are the features of the product family The traditional BOMs of these final products are depicted in Figure 4 The planning modules and their BOMs which can result after modularization are depicted in Figure 5

It may seem that modularizing the BOMs as in Figure 5 does not establish a major enhancement because the number of resulting planning modules equals the number of final products in the initial situation However, in more complex product families with n features each of which has 2 options, modularizing the traditional BOM may result in the reduction from 2^n ($2 \times 2 \times 2 \times$) different final products which must be planned traditionally to $2n + 1$ planning modules which must be planned after modularization That is, one planning module for each option ($n \times 2$) and one planning module containing component products which are common (1) to all final products in that product family This under the assumption that all component products are selected through a single option In that case substantial simplification is realized

Often an additional BOM structure is defined for the product family, to support the translation of the forecast volume for the product family as a whole into the volumes for the separate planning modules This kind of BOM structure is generally called the percentage BOM Some standard software systems allow features to be explicitly represented in the percentage BOM, in order to facilitate the selection of planning modules or options in the order entry process Figure 6 shows the percentage BOM of the product family desk-lamps in the case features are represented in the structure An order-

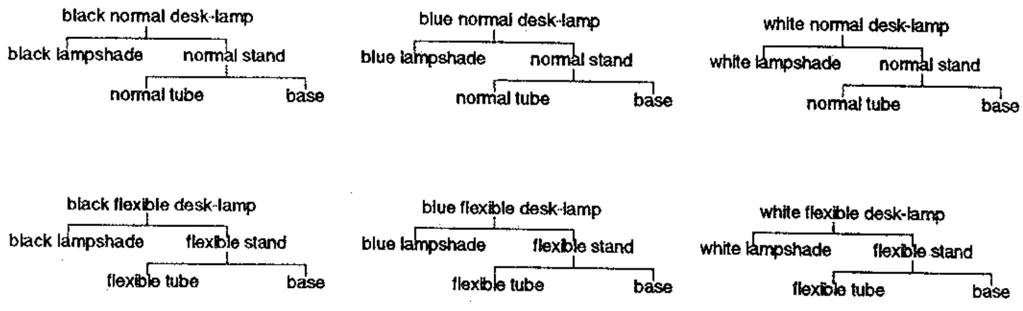


Figure 4 Traditional BOMs of six different variants of desk-lamp

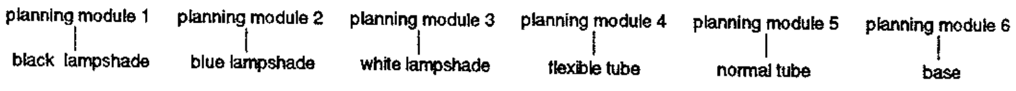


Figure 5 Alternative modular BOMs for desk-lamps

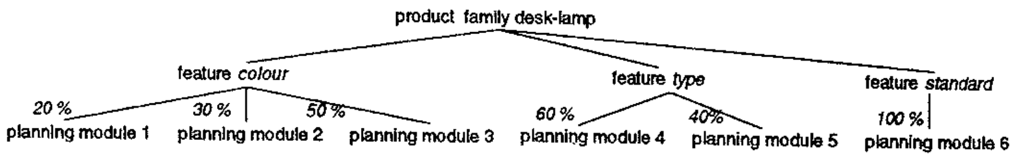


Figure 6 The percentage BOM of the product family desk-lamp, including features

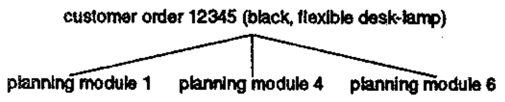


Figure 7 BOM of a particular customer order specifying a black, flexible desk-lamp.

dependent BOM is established in the process of selecting the required planning modules or options. For example, when a black, flexible desk-lamp is required, one appropriate planning module or option must be selected for each of the features, and a single-level BOM is created for the customer order in question (see Figure 7).

4 2 *Implicit assumptions in modularization*

The assumption that each planning module equals an option is not the only assumption which implicitly underlies the technique of modularizing BOMs. It is the first of the following four assumptions generally made.

Assumption 1 Planning modules and options coincide It is assumed that planning modules can be defined in such a way that they are significant units to the market, i e they are suitable for forecasting (Wijngaard 1987) and for order entry (Kneppelt, 1984) In other words it is assumed that each planning module is in fact an option and vice versa. In that case the BOM of a planning module defines the full material contents which are required to realize one particular option (Mather 1982) [1]

Assumption 2 Options of different features are independent[2] It is assumed that in forecasting and order entry, an option can be selected for a feature without any implications for the selection of options for other features If this were not assumed, options could neither be forecast nor selected in the order entry process independently of each other Note that in that case the question should be raised whether separate options are significant to the market in the first place

Assumption 3 Options of the same feature do not have lower-level products in common[2] Wortmann (1987) has pointed out that from a material planning point of view it is important that planning modules can be created which have few or no component products in common If overplanning is applied to options within the same feature, the products which are common to these options will be over planned for each of these options, leading to more safety stock than is actually required

Assumption 4 Assembly BOMs can be reconstructed It is assumed that no additional information is required for assembling a final product from the component products defined in the planning modules of that particular product, or that techniques are available to compensate for the loss of information caused by abolishing higher level BOMs in the modularization process (Vollmann *et al* 1988, Kneppelt 1984, Orlicky *et al* 1972, Sari 1981) Consider for example the modular structure of Figure 5 If a final product has been specified, it is known that a base, tube, and lampshade must be assembled into a final product However, the information represented by the traditional assembly BOMs, that the base and tube should be assembled first, is no longer available

The problem of the enormous number of BOMs can only partly be solved by modularizing BOMs The BOMs of *final* products may be removed without any serious consequences because they are in fact reconstructed as the planning modules are selected However as lower-level BOMs are removed information which is relevant to the assembly process may be lost and cannot be reconstructed easily For a simple product such as a desk-lamp this will not be a problem, but it will become a problem for more complex products such as cars and complex machinery Orlicky *et al* (1972) noticed the problem and proposed retaining the BOMs of sub-assemblies such as the stand The assembly BOMs which must be retained may still be far too numerous to handle The MRP literature does not provide satisfactory solutions for supporting the creation of an assembly BOM after a unique product description in terms of options has been drafted

4 3 *Conclusions*

The success of the technique of modularizing BOMs relies heavily on the extent to which a number of assumptions concerning the way in which product families, planning modules, and features and options can be defined are met.

As a consequence of the increasing variety and complexity of products, these assumptions will be less often valid It is often impossible to create planning modules which can play the role of both recognizable options for identifying products and of defining planning modules for the BOMs of these products Any pragmatic redefinition and restructuring of product families, features, options and BOMs in order to meet above-mentioned assumptions again, in fact, takes us back to where we

[1] If the planning module with common component products is regarded as an option

[2] Of course already under the assumption that an option equals a planning module

started. an enormous number of planning modules or options which have to be defined explicitly by a part number and a BOM defining the material contents (van Veen 1992) Consequently there is a great need for concepts which allow options and planning modules to be defined separately This requires the possibility of defining relationships between options and BOMs and options mutually

Far too little attention has been paid to the problem of creating and maintaining assembly BOMs to support final assembly. This type of support will become more and more important due to the fact that increasing product variety will force manufacturers to perform more and more activities customer order driven as opposed to forecast driven In other words, the number of the processes beyond the customer order decoupling point will tend to increase, which will increase the need for means of control (van Veen 1992)

In the next section a functional architecture is presented for more flexible product specification and BOM processing systems, so-called generative BOM processing systems

5. An architecture for a generative bill of material processing system

For generative BOM processing systems the concepts *item* and *product variant* need to be introduced Roughly speaking an item represents a set of similar products, which can be described by parameters These parameters are similar to the features introduced earlier in this paper, but we want to give parameters a more specific meaning. they constitute features which are significant in the market and can be used during order entry. A product variant represents a particular product The products represented by the same product variant are mutually exchangeable A product variant can be identified by a set of parameter values which represent the product characteristics of that particular product variant Such a set of parameter values is called a *specification.* An item is *a set of* different product variants, which have the same value for at least one parameter A specification S is valid against an item X if X contains a product variant which is uniquely identified by S. If a valid specification S is assigned to an item X, that item will be called *fully specified*

In the previous section, it was concluded that it is often impossible to define modules which are suitable for both identifying a final product and defining the BOM of that final product at the same time Therefore, two separate processes are suggested, namely.

(1) The product specification process in which a product variant is merely *identified* by means of parameter values

(2) The BOM generation process in which a BOM is generated for a product variant which is identified by means of parameter values

The distinction of these processes is used for a basic architecture for a new kind of BOM processing system In this architecture two core subsystems are distinguished, namely (see Figure 8):

- The *product specification system* (PSS), the primary function of this system is to evaluate whether a specification S is valid or invalid against a particular item A
- The bill of material generating system (BGS), the primary function of this system is to generate a BOM, given an item A and a valid specification against A The BGS should not only support the generation of a single-level BOM for a fully specified item, but if such exists also the generation of a multi-level BOM.

The PSS relies on the definition of items and for each item on the definition of the set of specifications that are valid for that item, the so-called set-description of that item The data which represent items and their set-descriptions will be called *product specification data* The BGS relies on a so-called source BOM This can be a single-level or a multi-level BOM The *source BOM* of an item A is the set of gozinto-relationships in which A is the parent item In the source BOM, A and its component items may contain one *or more* product variants The multi-level BOM which is obtained for a fully specified item A, by selecting and/or specifying gozinto-relationships and items from the multi-level source BOM

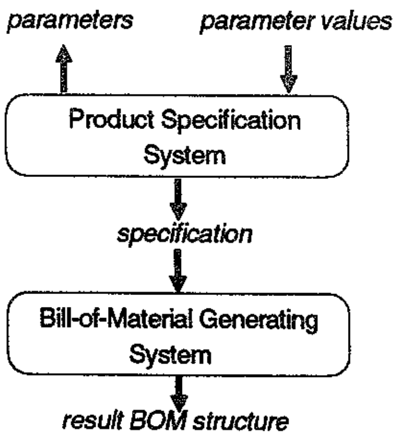

parameters
parameter values
Product Specification System
specification
Bill-of-Material Generating System
result BOM structure

Figure 8 An architecture for a new kind of BOM-processing and order entry system

of A, is called the multi-level result BOM The multi-level result BOM of the fully specified item A unambiguously represents the product structure of the product variant which is identified by the fully specified item A. It can be manually derived from a multi-level source BOM but in the case of a BOM generating system it is assumed that the multi-level source BOM holds data which allows multi-level result BOMs to be automatically generated given an item and a valid specification.

The above introduction of many concepts will now be illustrated by its most simple implementation: the so called variant BOM.

6. A generative bill of material concept: the variant BOM

Once a product variant has been identified by its item and a valid specification, it should be possible to obtain the BOM structure of that product variant unambiguously and immediately. In the traditional BOM concept this process is trivial: once the part number is known all BOM relationships which refer to that part number as the parent can be retrieved In the case of modular BOMs it is assumed that a product variant is identified by composing a single-level BOM consisting of planning modules (see Section 4 1). In the generative BOM concept it is assumed that product variants can be identified by a specification and that complex dependencies between parameter values of these specifications and gozinto-relationships exist in the source BOM. Based on this information a BOM generating system can generate a multi-level result BOM Schönsleben (1985) describes as one of the first authors a BOM processing system which allows a generative multi-level source BOM to be defined. This systems is called 'Variantengenerator VAR'.

According to Schönsleben, the objective of this BOM processing system is to achieve the optimum in recording and retrieving BOMs and routeings of large sets of product variants Schönsleben assumes that a range of product variants can be defined by a set of parameter values such that.

(1) the parameters are mutually independent, i e no constraints on combinations of parameter values exist[3], and
(2) by determining the values of the parameters for a range of product variants the required component product variants can be selected unambiguously

This BOM generating system VAR is based on a particular kind of generative BOM concept. the *variant BOM concept* The generative source BOMs can be created as a result of an extension of the database model of the traditional BOM concept In the variant BOM concept the source BOM of an item X is a set of gozinto-relationships in which X is the parent item However, this set is partitioned into explicitly defined subsets Each of these subsets may have one or more members We will call such a subset a *cluster* The idea is that each result BOM should consist of precisely one element out of each cluster. In the terminology of the variant BOM concept, a gozinto-relationship in the source BOM is usually called a *BOM relationship variant* The entire set of BOM relationship variants which have the parent item X, is called the variant BOM of X (the equivalent term of the source BOM of X) From here on, a BOM relationship variant will be shortened to *BR-variant*

As was mentioned, a result BOM is also a set of BR-variants However a specific requirement on the result BOM is that each cluster only has one member The principle of the BOM generation process consists of selecting none or one BR-variant from each cluster in the (multi-level) source BOM to constitute the (multi-level) result BOM The variant BOM concept described by Schönsleben may be called a generative BOM concept because, for each cluster in the source BOM-structure, a set of rules can be defined which express dependencies between the BR-variants of that cluster and parameter values, such that the required BR-variants can be selected automatically, given a set of parameter values These rules are usually called *conditions* The set of conditions for one cluster is in fact a decision table Note that expressing dependencies between parameter values and BR-variants is not limited to a single-level BOM It can also be applied in a multi-level BOM Generating a multi-level result BOM consists of having BR-variants automatically selected from the multi-level source BOM given an item and a set of parameter values When a multi-level result BOM must be generated, each selected BR-variant refers to a component item For each of these component items again one or more clusters may be defined, and again none or one BR-variant can be selected from these clusters, until the lowest-level items are reached. Notice however that the same set of parameter values is applied for all clusters, regardless of the level in the source BOM-structure

We will illustrate the generative multi-level source BOM according to the variant BOM concept for the six desk-lamp product variants in Figure 4 Figure 9 describes the generative multi-level source BOM for the

[3] Schönsleben concentrated on the bill of material generating system However as we have discussed, generally the product specification system cannot be disregarded As we have discussed, it must be possible to define incompatible parameter values (e g because of technical or legal constraints) of product family items Such incompatibilities are often called constraints

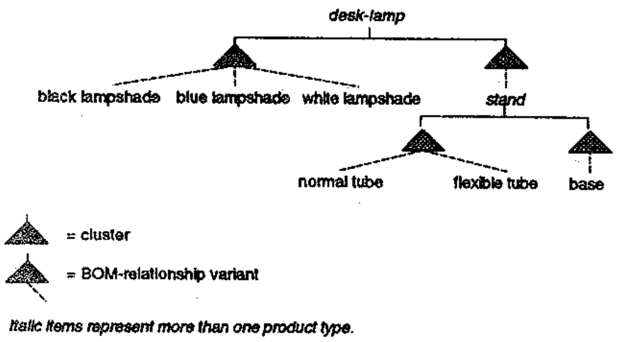


Figure 9 Multi-level source BOM for six different desk-lamp product variants

item representing the six desk-lamp product variants according to the variant BOM concept.[4] Table 3 shows the BOM relationships and conditions of the source BOMs The combination of the parent item and the sequence number identifies a cluster.

Schönsleben also created a facility to calculate the quantity/per by means of a predefined formula in terms of the parameter values of the top-level final product item This is particularly suitable when a large number of BR-variants between the same parent and component part number exist as a result of different values for the quantity/per attribute. In this case, the members of a cluster are no longer explicitly enumerated. That is, not each variant has to be explicitly redefined. In this paper we will not discuss this aspect in further detail.

The typical variant BOM concept carries a number of shortcomings, which predominantly arise from the fact that parameters, parameter values and constraints cannot be exclusively associated with an item at an arbitrary level in the product structure In the typical variant BOM concept it has been chosen only to allow set-descriptions to be maintained for product families By this design choice a simple solution emerges, but also a number of difficulties arise. The advantage is that a BOM generating system which allows multi-level result BOMs to be generated for *final* product variants can be realized with little effort However, the concept may easily lead to data redundancy and many problems with engineering changes These shortcomings will be analysed in the next section.

In a second paper titled 'New developments in generative BOM processing systems' an enhancement of the variant BOM concept called the *generic BOM concept* will be introduced In this concept it will be allowed to define set-descriptions for arbitrary items. However this will lead to some other complications which require a number of new concepts to be developed

[4] A decision table could be defined for each triangle which has multiple lower-level variants In Figure 9 for clarity these decision tables are not shown

7. Limitations of the variant BOM concept

7.1 *The problem of representing product variety at lower levels in the product structure in the variant bill of material concept*

In the variant BOM concept, product specification data (i e. parameters, parameter values and constraints) are exclusively related to product family items In other words, the product specification data implicitly define a set of final product variants. Product variants in a lower-level item X cannot be described in terms of their own characteristic parameters and parameter values A lower-level item X *can* represent a set of more than one product variant because different multi-level result BOMs can be generated from the multi-level source BOM of X The number of different product variants represented by X is in fact determined by the number of different multi-level

Table 3 BOM relationship variants for the generative mutli-level BOMs of the six desk-lamps

BOM relationship variant					
Cluster					
parent	seq	variant	qty/per	component	condition
desk-lamp	10	1	1	black lampshade	(colour, black)
desk-lamp	10	2	1	blue lampshade	(colour, blue)
desk-lamp	10	3	1	white lampshade	(colour, white)
desk-lamp	20	1	1	stand	
stand	10	1	1	flexible tube	(type, flexible)
stand	10	2	1	standard tube	(type, standard)
stand	20	1	1	base	

result BOMs which can be generated from the multi-level source BOM of X. Consider for example the lower-level item *stand* in the source BOM-structure of the *desk-lamp* from Figure 9. In this source BOM it is assumed that the item *stand* represents two product variants because two different result BOMs can be generated from this source BOM. Usually not all BR-variants from different clusters can be combined randomly. Which result BOMs are released, is determined by the different specifications which are valid against a product family item set-description.

However, a first problem arises because a set-description can only be maintained for a product family. Assume $n+1$ product variants have been released for an item X. But only product variants $1, \ldots, n$ are applied in one or more final product variants. The problem is how to represent the fact that the product variant $n+1$ has actually been released. The set-description which allows the specification for which the multi-level result BOM of product variant $n+1$ would be generated, cannot be associated with an entity of the type product family, because that would implicitly allow final product variants to be specified which have not been released (i.e. final product variants which have product variant $n+1$ as a component). However this set-description can also not be uniquely associated with X, because this is not allowed in the typical variant BOM concept. In other words, it is impossible to define a set of product variants at a lower level in the source BOM-structure if not all of these product variants are applied in a final product variant. In Appendix 3 this problem is illustrated in detail with an example.

For many applications, the shortcoming of not being able to represent sets of lower-level product variants independently of sets of final product variants is serious. In particular if ranges of lower-level product variants are released by engineering at the same time, but not all of these product variants are part of a final product variant (e.g. in the case of time phased marketing release of final product variants). The product variants not yet applied in a final product variant cannot be represented by an item and its source BOM. Assume, for example, a range of similar generator product variants (a major component in a medical diagnostic system) has been developed and released and we would like to represent the BOMs of the entire set of generators by one source BOM structure. This will be a problem if not all of these generators are a component of a final product variant, e.g. a medical diagnostic system. The range of generators cannot be defined as a set by a product family *generator*, because—according to the limitation chosen in the variant BOM concept—it could then no longer appear as a component item in the source BOM of a product family item *medical diagnostic system*. No result BOM can be generated for the generators which *have been* **engineered and released, but which** *are not (yet) applied* in a medical diagnostic system.

The obvious solution is to allow independent set-descriptions to be maintained for arbitrary items, or in the terminology of the variant BOM concept, to allow product family items to be defined at arbitrary levels in the generative source BOM-structure. However, as mentioned already, this requires additional concepts to be developed, in order to cope with difficulties which will arise (van Veen and Wortmann 1992).

7.2 *Data redundancy in parameters, parameter values and constraints in the variant bill of material concept*

The source BOMs may contain considerable data redundancy in parameters, parameter values and constraints because some of this data which is exclusively related to lower-level product variants can only be explicitly defined for end items. A certain combination of parameter values may be invalid for many product families, due to technical restrictions on a particular type of sub-assembly. Consider the example of medical diagnostic systems which consist of many components, including a generator. Some parameters, parameter values and constraints may solely be required for determining a required generator product variant. Independent of the medical system in which a generator is applied, constraints caused by technical incompatibilities with regard to generators may have to be met. Because in the variant BOM concept only set-descriptions to be maintained for product families are allowed, the parameters, parameter values and constraints which have an isolated effect on lower-level item A must be defined for each product family in which A is applied. In Figure 13 of Appendix 3 for example, the constraint on the combination of parameter values $\{(p_1, v_{11}), (p_2, v_{21})\}$ may be required because of a technical infeasibility of a product variant of A (e.g. components from item X and item P are technically incompatible). In the variant BOM concept this restriction must be defined for both X_1, and X_2. This type of data redundancy increases with number of items for which parameters and parameter values have an isolated effect and the number of product family items in which these items are applied. Such data redundancy is not merely a problem of computer capacity. It must also be associated with problems in data maintenance. Modifications required because of changes in one or more generators may have to be implemented in the set-descriptions of many different end items instead of only for the item representing the generator product variants.

Another form of data redundancy which occurs in

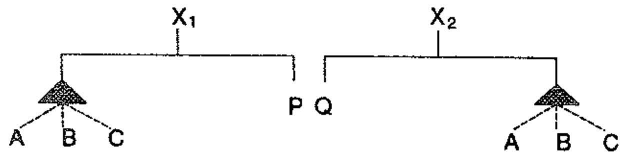


Figure 10 Data redundancy in the source bill-of-material structure

source BOMs based on the variant BOM concept arises from the fact that sets of alternative component items have to be defined each time they appear in a cluster as alternative component items Figure 10 shows an example of this form of data redundancy for three items A, B, and C The set of alternative items must be defined once for X_1 and once for X_2 The severity of this form of data redundancy increases as particular sets of the same items are applied many times, in different clusters

A pragmatic solution consists of defining an artificial parent item with a source BOM consisting of one cluster only, which specifies the set of alternative items as alternative component items In that case the artificial item specifying a set of alternative components has to be defined only once, after which it can be applied in as many higher level BOMs as desired In the example of Figure 11 an artificial parent item A' is defined with a source BOM specifying A, B, and C as alternative component items.

As mentioned, A and its lower-level source BOM are only defined once and can be applied in as many higher level cluster as required In the result BOM structures generated from the source BOMs, this artificial parent item will need to be suppressed and will not appear as an additional level in the multi-level result BOMs This solution is in fact a step towards the generic BOM concept (van Veen and Wortmann 1992)

Finally, a form of redundancy may occur if an item A is applied in more than one product family and the sets of product variants defined by these product families are defined by different sets of parameters and parameter values The redundancy occurs in the decision tables which control the selection of a BR-variant for a cluster of A As the generation process may start from different product families, different parameter values may be used

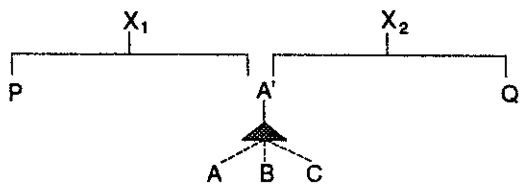


Figure 11 Defining an artificial parent item in order to represent a set

Table 4

Parent item	Sequence number	BR-variant	Decision table rule	Parameter value
A	10	01	1	v_{11}
A	10	02	1	v_{12}
A	20	01	1	v_{21}
A	20	02	1	v_{22}

to drive the generation process Additional decision table rules will need to be defined in order to be able to select a BR-variant This is merely because the parent item A is applied in different product families, which are characterized by different parameters and parameter values This can be envisaged by Table 4 which shows the decision table rules of the source BOM structure of a parent-item A

Assume that a new product family, X_3 is defined with the following set-description.

$$\{p_3 \varepsilon \{v_{31}, v_{32}\}$$
$$p_4 \varepsilon \{v_{41}, v_{42}\} \mid$$
$$\text{NOT } ((p_3, v_{31}) \text{ AND } (p_4, v_{41}))$$
$$\text{NOT } ((p_3, v_{32}) \text{ AND } (p_4, v_{42}))\}$$

Then for each BR-variant in the BOM of A, an additional decision table rule must be defined to allow for the generation of the same BOMs for A, for the case where A is applied in the product family X_3, instead of X_1, or X_2 Table 5 shows the decision table rules required for the source BOM structure of A in that case A negative side-effect of this type of data redundancy is that it becomes increasingly difficult to determine upon which parameters a lower-level item is actually dependent The decision tables in the BOM of such an item may become very complex and difficult to comprehend It should be noted that this type of redundancy can be avoided if it can be achieved that for each parent item X, with a source BOM structure containing one or more decision tables, all end items in which X is applied, have

Table 5

Parent item	Sequence number	BR-variant	Decision table rule	Parameter value
A	10	01	1	v_{11}
A	10	01	2	v_{31}
A	10	02	1	v_{12}
A	10	02	2	v_{32}
A	20	01	1	v_{21}
A	20	01	2	v_{41}
A	20	02	1	v_{22}
A	20	02	2	v_{42}

set-descriptions in terms of the same parameters and parameter values

Notice also that in the variant BOM concept, modifications at product family level may have an impact on many *lower-level items and matching source BOMs* If a product family, parameter or parameter value is deleted, then it will probably be insufficient to modify only the set-description of the product family and the decision tables in its own source BOM It may also require modifications in the BOMs (in particular the decision table rules in these BOMs) at lower levels in the multi-level source BOM

8. Conclusions

Traditional methods for structuring BOMs do not perform satisfactorily if product variety increases strongly In the traditional MRP literature modularizing BOMs is suggested as a solution to this problem Although modular BOMs can support forecasting and master scheduling processes, they often do not support assembly and logistical control processes Some authors suggest retaining separate assembly BOMs for final products and if necessary sub-assemblies, but often the number of different product variants prohibits this solution For this purpose, the concept of generative BOMs is introduced

In this paper special attention was paid to the variant BOM concept being a particular kind of generative BOM Although the variant BOM concept may provide a relatively simple solution, especially for those situations in which product variety arises in the top-levels of the product structure, it also implies some principle complications of data maintenance These complications are due to data redundancy that may arise in parameters, parameter values, constraints and conditions This data redundancy in turn is the result of the fact that the variant BOM concept allows parameters and parameter values to be defined for product family items only

As mentioned before an enhanced concept is elaborated in our paper 'New developments in generative BOM-processing systems'

References

KNEPPELT, L R, 1984, Product structuring considerations for master production scheduling *Production and Inventory Management*, first quarter, pp 83–99

MATHER, H, 1982, *Bills of Materials, Recipes and Formulations*, (Wright Publishing Company, Atlanta)

ORLICKY, J A, 1975, *Material Requirements Planning* (McGraw-Hill, New York)

ORLICKY, J A, PLOSSL, G W, and WIGHT, O W, 1972, Structuring the bill-of-material for MRP *Production and Inventory Management*, **13** (4), 19–42

SARI, J F, 1981, *The MPS and the Bill of Material Go Hand-in-Hand* (Richard C Ling, Winston Salem)

SCHÖNSLEBEN, P, 1985, *Flexible Produktionsplanung und Steuerung mit dem Computer* (CW-Publickationen, München)

VEEN, E A VAN, and WORTMANN, J C, 1987, Generic bills-of-material in assemble-to-order manufacturing *International Journal of Production Research*, **25** (11), 1645–1658

VEEN, E A VAN, 1992, *Modelling product structures by generic bills-of-material* (Elsevier Science Publishers B V, Amsterdam)

VEEN, E A VAN, and WORTMANN, J C, 1992, New developments in generative BOM-processing systems *Production Planning and Control*, **3**, 327–335

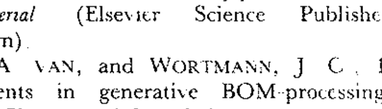
VOLLMANN, T E, BERRY, W L, and WHYBARK, D C, 1988, *Manufacturing Planning and Control Systems* (Dow Jones-Irwin, Illinois)

WIJNGAARD, J, 1987, Production control in a consumer electronics factory *Engineering Costs and Production Economics*, **12**, pp 165–174

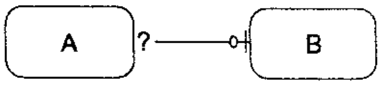
WORTMANN, J C, 1987, Information systems for assemble-to-order production. an application *Engineering Costs and Production Economics*, **12**, pp 187–196

Appendix 1. Notation conventions in data structure diagrams

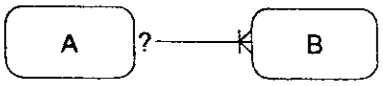
1 The data structure relationship in Figure A1 1 represents the fact that for each entity *a* of the type *A* *precisely one* entity *b* of the type *B* exists which refers to the entity *a*



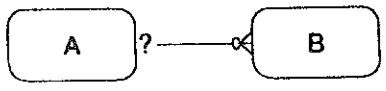
2 The data structure relationship in Figure A1 2 represents the fact that for each entity *a* of the type *A*, *none or one* entity *b* of the type *B* exists which refers to the entity *a*



3 The data structure relationship in Figure A1 3 represents the fact that for each entity *a* of the type *A* *at least one (thus potentially more than)* entity of the type *B* exists which refers to the entity *a*



4 The data structure relationship in Figure A1 4 represents the fact that for each entity *a* of the type *A* *none, one or many entities* of the type *B* exist which refer to the entity *a*



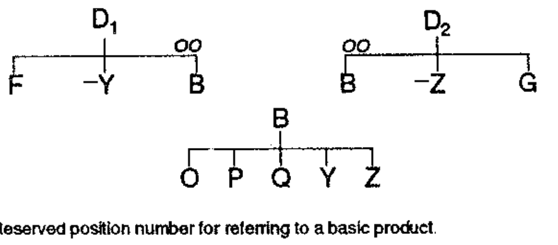
If because of one or more data structure relationships, given an entity a of the type A one or more entities b_1, , of the type B can be found unambiguously, then a is associated with b_1, ,b_n Also, each $b_i \in \{b_1, ,b_n\}$ is associated with a

Appendix 2. The add-and-delete technique

The add-and-delete technique has been developed with the objective to reduce the total number of BOM relationships which is required to define the BOMs of a range of products This should facilitate BOM maintenance and lead to a reduction of the required computer storage capacity which in turn would improve the performance of the PCIS In the add-and-delete technique, a *basic product B* is distinguished for which a part number and a complete BOM are defined Other products which are technically *derivatives* from B are also explicitly defined by a part number However the BOM of such a derivative merely lists the *differences* compared to the BOM of B Consider a basic product B and a derivative product from B, D_1 Products which are a component of B but which are not a component of D_1, are specified in the BOM of D, with a *negative quantity/per* The fact that D_1, is a derivative of B is represented by referring to B as if it were a component product (see Figure 12) Usually a particular position number (e g 00) in the BOM is reserved for reference to a basic product In that way, it can be recognized during a BOM explosion of a derivative, that a particular BOM relationship refers to the basic product This product may then be disregarded (if such is desired) in the output of the BOM explosion

Mather (1982) states that this technique may be useful from the viewpoint of BOM maintenance However, he also discusses some disadvantages from the viewpoint of material planning The most fundamental disadvantage is, according to Mather (1982) the fact that add-and-delete BOMs do not allow overplanning on specific products, i e derivatives only If requirements on derivatives are overplanned, then the requirements on the basic product will also be Another important disadvantage mentioned by Mather concerns the number of part numbers which must be scheduled Add-and-delete BOMs do not reduce the number of (final) product part numbers which must be master scheduled For a more detailed discussion of material planning problems which may arise if the add-and-delete technique is applied refer to Mather (1982)

The assumption that the add-and-delete technique facilitates BOM maintenance may be true in the case of initial BOM-definition and if the majority of the subsequent engineering changes has an isolated impact on the basic product, i e requires no separate modification in the derivatives of the basic product Sometimes the initial BOMs of basic products degenerate into artificial sets of BOM relationships which are no longer related to recognizable products but only exist because they specify a number of components which are common to many other products In these cases the BOM structures may become very difficult to comprehend and the add-and-delete technique may become a burden Another problem in data maintenance is that inconsistencies may easily arise Very complex software is required to prevent inconsistencies Consider for example Figure 12; BOM maintenance software must at all times prevent the removal of component product Y from the BOM of B, if it is still specified in the BOM of D_1 with a negative quantity/per Recall that the add-and-delete technique is primarily focused on reducing the number of BOM relationships which must be stored in a database As data storage facilities become cheaper and the functionality of BOM processors to manipulate BOMs (such as copying facilities, increases, the advantages of the add-and-delete technique become less important From the viewpoint of BOM representation in the case of very large product variety, the add-and-delete technique is not a satisfactory solution It still requires that each product variant is defined explicitly by a separate part number and that a BOM must still be defined for each part number (although it may contain fewer BOM-relationships) In the case of millions of products even the add-and-delete technique will prove to be of little use



OO = Reserved position number for referring to a basic product.

Figure 12 The add-and-delete technique

Appendix 3. Representing product variety at lower levels in the variant BOM

Figure 13 illustrates that it is impossible to determine whether an assumed product variant f, described by the multi-level result BOM generated for the specification $\{(p_1, v_{11}), (p_2, v_{21})\}$[5] has been released in A, if that

[5] p_i: parameter i, v_{ij}: value j for parameter i

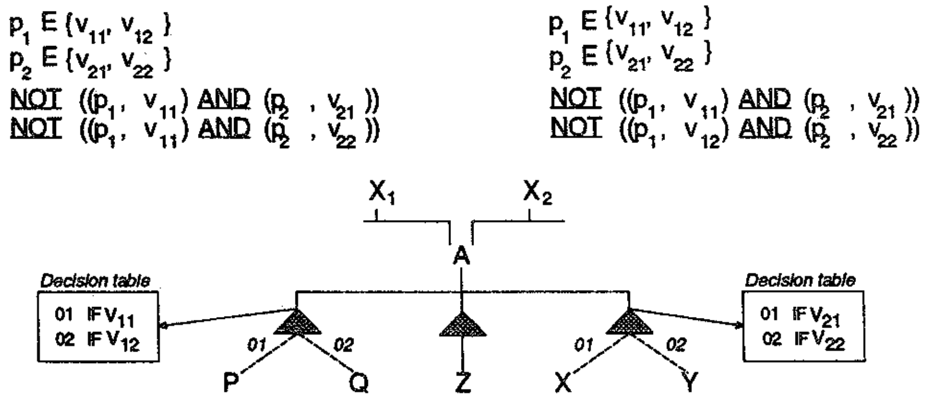

$p_1 \in \{v_{11}, v_{12}\}$
$p_2 \in \{v_{21}, v_{22}\}$
NOT $((p_1, v_{11})$ AND $(p_2, v_{21}))$
NOT $((p_1, v_{11})$ AND $(p_2, v_{22}))$
$p_1 \in \{v_{11}, v_{12}\}$
$p_2 \in \{v_{21}, v_{22}\}$
NOT $((p_1, v_{11})$ AND $(p_2, v_{21}))$
NOT $((p_1, v_{12})$ AND $(p_2, v_{22}))$
X_1
X_2
A
Decision table
01 IF V_{11}
02 IF V_{12}
Decision table
01 IF V_{21}
02 IF V_{22}
01
02
01
02
P
Q
Z
X
Y

Figure 13 An example of a multi-level source BOM according to the variant bill-of-material concept

product variant is not applied in a final product variant The figure depicts two product families X_1 and X_2 with a common component item A which has a lower-level source BOM Disregarding the product families in which A is applied, four different result BOMs could be derived from the source BOM of A[6]. One of these four is the BOM which specifies the component items P and X This result BOM will be generated for the specification $\{(p_1, v_{11}), (p_2, v_{21})\}$. We are interested in the question, whether this result BOM has or has not been released (the latter, for example, because the components represented by P and X are incompatible)

The only possibility of determining whether a product variant with the result BOM structure in question has been engineered and released, is to find out whether this BOM structure can be generated for at fewest one final product variant In that case the product variant will certainly have been engineered and released Otherwise (i e if that result BOM structure cannot be generated as a part of the result BOM structure of a final product variant) the lower-level product variant is not applied in a final product variant and the question remains unanswered

Examining the specification $\{(p_1, v_{11}), (p_2, v_{21})\}$ it shows that there is no final product variant identified by this specification It is not allowed to conclude that a product variant f consisting of the items X and P has been released, but that it is just not applied in a final product variant (yet): a technical incompatibility of product variants from X and P may be the very reason that there is no final product variant which consists of a product variant f. On the other hand the conclusion that a product variant f is not released, because it is not applied in a final product variant would only be justified if no product variants are released which are not applied in at least one final product variant This is of course a very far-reaching assumption.

[6] Assuming that, at fewest, one BR-variant must be selected for a cluster