New developments in generative BOM processing systems

PRODUCTION PLANNING & CONTROL, 1992, VOL 3, NO 3, 327–335

New developments in generative BOM processing systems

E A VAN VEEN and J C WORTMANN

Abstract The principle of generative bill-of-material (BOM) processing systems is that different BOMs belonging to different product variants can be represented by a single, so-called source BOM The BOM processing systems comprise additional data structures which hold information on the relationships between product characteristics of parent product variants and component product variants, and on the relationships between characteristics of a parent product variant and its BOM data These relationships allow the automatic generation of the individual BOM of each represented product variant There are several alternative ways of implementing a generative BOM processing system The oldest concept known is the variant BOM concept This concept provides a relatively simple solution to deal with large varieties of final product variants However the concept has a number of drawbacks such as the representation of product variety at lower levels in the product structure and data redundancy which hampers data maintenance In this paper an improved concept for generative BOM processing systems is introduced and described. the generic BOM concept The generic BOM concept does not focus on representing final product variants only, but takes a broader view towards representing any range of product variants at any level in the product structure This starting point solves a number of drawbacks implied by the variant BOM concept but it also requires new definitions of BOM relationships and the introduction of new data structures to support the generation of individual BOMs

1. Introduction

In our previous paper 'Generative bill-of-material processing systems' we discussed the concept of generative bill-of-material (BOM) processing systems in general and the variant BOM concept in particular It was concluded that as a result of design choices made in the variant BOM concept, a simple solution for representing BOMs for wide product varieties emerged However these design choices also imply some drawbacks, which were extensively discussed in the previous paper In this paper we will introduce an improved concept. the generic BOM concept In Section 2 we will define some key concepts in generative BOM processing systems In Section

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

3, we will introduce the main demands to be imposed on a new generative BOM concept. In Section 4 the generic BOM concept will be introduced Section 5 summarizes the main conclusions

2. Definitions

Key concepts in the generative BOM concepts are *product variant*, *parameter value* and *item* A product variant represents a particular kind of product All products represented by a product variant are assumed to be mutually exchangeable A product variant is 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 one, or more different product variants It is *generic* if it contains more than one product variant It is *specific* if it contains precisely one product variant In this paper, product variants will be denoted by lower case letters, items by upper case letters

A specification S is called *valid* against an item X if X contains a product variant which is uniquely identified by S X is then called a *fully specified* item The *set-description* of an item X defines the set of valid specifications for X, in terms of parameters, parameter values and (in)compatible combinations of parameter values

In terms of generating a BOM for a fully specified item, the BOM from which the generation process starts is called the *source BOM*. The BOM which results after the generation process is called the *result BOM*

3. Demands on a generative bill-of-material concept

The objective of generative BOM concepts is to avoid the explicit definition of individual product variants by part numbers and to avoid the explicit definition of each BOM of a product variant The basic idea is that information which had to be repeated for each individual product variant in traditional BOM concepts, only has to be recorded once in the generative BOM concepts

The drawbacks of the variant BOM concept have led us to the formulation of six demands to be imposed on a new generative BOM concept These demands are discussed in detail by Van Veen (1992) In this paper we will focus on the following three of these six demands.

First, in a generative BOM-concept, the explicit definition of sets of product variants should not be limited to final product variants only. It should be possible to define sets of product variants at lower levels in the product structure These sets should be defined in terms of parameters and parameter values which specifically apply to these product variants For example, it should be possible to define a set of engines, independently of which engines are a component in a final product variant (e g cars) and which are not Thus the set of engines may comprise one or more engines which are not components in a higher level product variant

Second, in the variant BOM concept the set of product variants represented by a lower-level item is in fact determined by the set of different result BOMs that can be generated for that item In a new concept it should be possible to define sets of product variants which do not have a lower-level source BOM Within these sets it should be possible to identify lower-level product variants by means of a specification This for example applies to raw materials or purchased products

Third, a new generative BOM concept should make it possible to represent the *general product structure* of a set of similar product variants * But it should also be possible to automatically generate the multi-level result BOM of each of the individual product variants, given the specification of the product variant in question It may be desirable to define one source multi-level BOM representing the general product structure of a large set of similar cars In the end however, the multi-level BOM of each individual car must be made available immediately if that car is to be manufactured

To meet these demands, other design choices than that of the variant BOM concept must be made In the subsequent sections we will introduce the generic BOM concept This concept meets the demands above, but it also requires a new definition of the concept *gozinto-relationship* and new data structures to represent product structure data called conversion rules

4. The generic bill-of-material concept

The basic design choice in the generic BOM concept is to allow the definition of a set of product variants by means of an item and a set-description at any level in the multi-level BOM This is opposed to the variant BOM concept in which such items can only be top-level items

In the generic BOM concept we can distinguish between *source-items* and *result-items* A result-item is generated from a source-item A source-item is defined by an item-number and a set-description. A result-item is an item which has been assigned a specification during the generation process and which refers to the source-item it

* The term 'general product structure' will be specified in further detail in Section 4 1

is generated from Hence, the (multi-level) *source* BOM of a source-item P is the (multi-level) BOM constituted by gozinto-relationships between source-items, which allows (multi-level) BOMs for individual product variants from P to be generated

The (multi-level) result BOM for an individual product variant p, which is a member of item P, is generated as follows.

Specification S identifies p within P S is assigned to P which generates the result-item P' P' is a result-item which has one member only, i e the product variant p The source BOM of P is exploded to find its component items For each of these component items again a valid specification must be obtained after which a result-item can be generated and so on, until result-items have been generated for the lowest-level source-items.

The next step is trivial. it consists of disregarding the fact that the gozinto-relationships in the multi-level result BOM are defined between fully specified result-items, and pretend that the gozinto-relationships are defined between the product variants identified by these fully specified result-items

In the following section we will explain the definition of gozinto-relationships in the generic BOM concept in greater detail In Section 4 2 we will show how traditional BOMs of product variants are to be represented in a generic source-BOM In Section 4 3 we will introduce a new concept called the conversion function which is required to determine the specifications to be assigned to lower level items during the generation of result BOMs We will show that this concept is required to meet the third demand from Section 3

4 1 *Gozinto-relationships in the generic bill-of-material concept*

Traditionally, gozinto-relationships are uniquely identified by the combination of a single key-attribute of the parent and a single key-attribute of the component (the parent part number and component part number), or by a single key-attribute of the parent and a sequence number for the gozinto-relationship (Van Veen 1992) In line with the objectives of the generic BOM concept the aim is to define items, being sets of product variants, and to define one source BOM for an item P, which applies to all product variants which are members of P Gozinto-relationships should therefore be defined between *sets* of product variants, i e between items, instead of between individual product variants

In the generic BOM concept, product variants are identified by a product specification which may consist of any number of parameter values Consequently gozinto-relationships can no longer refer to the single part number of the parent product variant and the single part number of the component product variant Also they can no longer be identified by their parent product variant and a sequence number. However, it will be assumed that items *are* identified by a single key-attribute, i e item number Therefore, in the generic BOM concept, a gozinto-relationship will be identified by the parent item number and a sequence number

It is important for the definition of the gozinto-relationship in the generic BOM concept to clarify the meaning of the term 'general product structure' The third demand stated that it should at all times be possible to unambiguously reconstruct the (multi-level) BOM of individual product variants But it also stated that a new generative BOM concept should make it possible to represent the general product structure of a set of product variants Two different definitions (each having a different meaning) can be associated with this term

According to the first definition a (multi-level) source BOM of an item G represents the general product structure of the product variants which are members of G, if the (multi-level) BOM of each product variant g which is a member of G can be generated from the (multi-level) source BOM of G This implies the following minimum requirement on the (multi-level) source BOM. if a gozinto-relationship exists between a parent product variant p and a component product variant c then a gozinto-relationship must exist between parent item P of which p is a member and component item C of which c is a member Note that it is not explicitly required that all other product variants which are members of P also have a component product variant which is a member of C

If this were the only requirement imposed on gozinto-relationships in the source BOM, given a set of gozinto-relationships between product variants, the definition of the gozinto-relationship would be the following.

> A gozinto-relationship between a parent item P and a component item C implies that *at fewest one* product variant which is a member of P has a component product variant which is a member of C

We can illustrate the consequence of this definition by the example of an item *CAR* and an item *ENGINE* If in the source BOM a gozinto-relationship existed between the parent item *CAR* and the component item *ENGINE*, this would only represent the fact that a car from item *CAR* may (or may not) have an engine of item *ENGINE* Maybe all cars of *CAR* have an engine of item *ENGINE* but then again maybe not Merely the fact that a product variant is a member of an item does not provide any definite information of its product structure (in terms of kinds of component product variants)

The second definition which can be assigned to the term general product structure imposes higher demands on the way items may be distinguished. But once the items have been distinguished, they provide much information on the product structure of their product variants By this definition it is assumed that the source BOM of item G represents the general product structure of the product variants of G, if for each g which is a member of G, it holds that.

- for each gozinto-relationship between a parent item P_s and a component item C_s in the multi-level source BOM of G, one gozinto-relationship between a parent item P_r and a component item C_r exists in the multi-level result BOM of g, and
- in which P_r is the result-item generated for P_s and C_r is the result-item generated for C_s

In other words if the multi-level source BOM of G contains an item X, than each product variant which is a member of G will have a component product variant which is a member of X In this case the definition of a gozinto-relationship is.

> A gozinto-relationship between parent item P and component item C implies that *each* product variant which is a member of P has a component product variant which is a member of C

In the example of the car, a gozinto-relationship between a parent item *CAR* and a component item *ENGINE* expresses the fact that each car of *CAR* has an engine of *ENGINE* By knowing that a product variant belongs to *CAR* it is immediately known that this product variant will have a component product variant of *ENGINE*.

By the first definition, no knowledge is obtained if it is known that a particular product variant p belongs to a source-item P The fact that P has a BOM does not provide any information on the BOM of p The particular product variant p could even be a monopart! By the second definition much more knowledge is obtained if it is known that a particular product variant p belongs to a source-item P It is immediately known from the source BOM of P which *kinds* of product variants will occur in the product structure of the product variant p will have a component product variant from each component source- item of P.

One should realize that additional knowledge obtained by using the second definition is only achieved at the cost of imposing higher demands on the way in which items must be distinguished given a set of gozinto-relationships. For example using the second definition, a

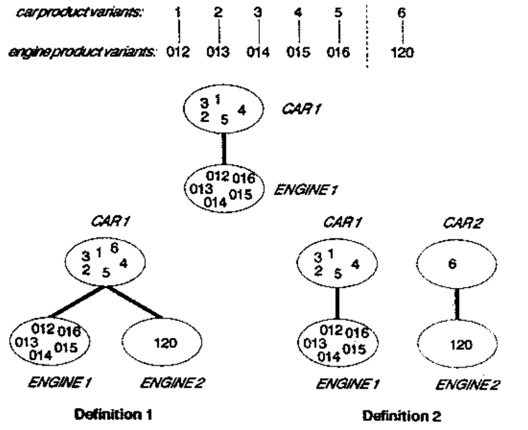

Figure 1 The definition of gozinto relationships

product variant *car 6* cannot be made a member of item *CAR* if it has a component product variant *engine 120* which is not a member of *ENGINE* (see Figure 1) In other words, separate items should then be defined for the product variant *car 6* and the product variant *engine 120*. Figure 1 depicts the consequences for both cases (i e the new situation according to definition 1 and definition 2) schematically

In this paper we will adopt the second meaning of the term 'general product structure' and consequently adopt the latter definition of the gozinto-relationship in the generic BOM concept Hegge (1992) adopts the first definition and develops a generic BOM processor based on that principle Given the definition of the gozinto-relationship, the subsequent sections explain how traditional BOMs can be represented by a (generic) source BOM

4 2 *Representing traditional bills-of-material in a source bill-of-material*

In practice items may be distinguished in many different ways At one extreme, few generic items can be distinguished, each generic item being a set of many product variants At the other extreme, many generic items can be distinguished, each consisting of only few product variants First, we will discuss an example of defining gozinto-relationships to constitute a source BOM for the extreme case that a separate item has been defined for every single product variant (i e only fully specified items are distinguished in the multi-level source BOM) From the viewpoint of the aim to simplify the representation of a product structure, this way of distinguishing items is of course senseless In this particular example, the source BOM does not improve insight into the product structure However, it is a good starting point to make clear the problems that arise if items consisting of *more than one* product variant are distinguished.

Consider the following product variants, parameters and values:

product variants	a, b, p, q, r
parameters	X, Y, I
sets of parameter values	X. $\{x_1, x_2\}$
	Y. $\{y_1, y_2\}$
	I. $\{123, 456, 789\}$

The product variants were initially identified by single characters a, b, p, q, and r. Recall that in the generic BOM concept, these single key-attributes may not exist and product variants are identified by a product specification (see Section 2). The product variants and their product specifications are listed in Table 1.

The traditional BOMs of these product variants are depicted in Figure 2. They consist of the gozinto-relationships listed in Table 2.

In this example the extreme case was chosen in which a separate item is distinguished for each individual product variant. This results in the definition of the items of Table 3. Notice the difference between Table 1 and Table 3. In Table 1 *one product specification* is associated with a product variant (denoted in lower case). In Table 3 a set-description i.e. a *set of product specifications* is associated with a source-item (denoted in upper case). Applying the definition of gozinto-relationships introduced in Section 4.1 to this set of source-items, the gozinto-relationships as specified in Table 4 will constitute the source BOM. Figure 3 depicts the source BOMs of A and B.

Table 1

Product variant	Product specification
a	$\{(X, x_1), (Y, y_1)\}$
b	$\{(X, x_1), (Y, y_2)\}$
p	$\{(I, 123)\}$
q	$\{(I, 456)\}$
r	$\{(I, 789)\}$

Figure 2 The bills-of-material of two different product variants

Table 2

Parent product variant	seq	Component product variant	Quantity/per
a	10	p	1
a	20	q	1
b	10	p	1
b	20	r	1

Table 3

Item	Set-description
A	$\{\{(X, x_1), (Y, y_1)\}\}$
B	$\{\{(X, x_1), (Y, y_2)\}\}$
P	$\{\{(I, 123)\}\}$
Q	$\{\{(I, 456)\}\}$
R	$\{\{(I, 789)\}\}$

Table 4

Parent item	seq	Component item	Quantity/per
A	10	P	1
A	20	Q	1
B	10	P	1
B	20	R	1

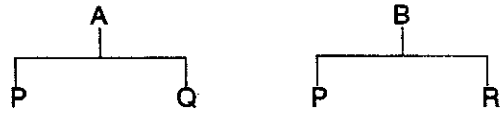

Figure 3 The bills-of-material of two items

Obviously, the way in which items have been distinguished in this example makes it very simple to generate the individual BOMs of the product variants. First the item to which the product variant in question belongs must be determined. The gozinto-relationships of that item provide the component items. Since in this example each component item consists of precisely one member, the product variant which is the component in the gozinto-relationship of the parent product variant is determined unambiguously. No further specification is required, each component item being already fully specified.

Because in this example all source-items are specific, the use of the generic BOM concept does not contribute to a better insight into the product structure of a range of product variants. This can only be the case if items have been distinguished in such a way that information which had to be repeated for each individual product variant in the traditional BOM concept, only has to be recorded once in the generic BOM concept, i.e. for the item of which these product variants are a member. Hence, a minimum requirement to obtain simplification is that some items consist of more than one product variant That is if the multi-level source BOM contains one or more generic items For example, if a number of different product variants representing cars are clustered into one item as are a number of different product variants representing engines, then the information that each car has one engine (traditionally represented by numerous gozinto-relationships between the individual product variants) can be represented by one gozinto-relationship between the items in question The subsequent section discusses a more useful way to distinguish items and the consequences for the generic BOM concept: the case in which generic items are applied

4 3 *Determining a component product variant: the conversion function*

The common situation is that items consist of *more than one* product variant It will be shown that additional information may have to be recorded in order to be able to unambiguously reconstruct the BOMs of individual product variants. We have already concluded that if a gozinto-relationship exists between product variant p and product variant c then, to be able to reconstruct this relationship in the result BOM, a gozinto relationship must exist between the source-item P of which p is a member and the source item C of which c is a member If an item P consists of many product variants p_1, , p_{i+1} and an item C consists of product variants c_1, , c_{j+1}, then according to the definition of the gozinto-relationship in a source BOM, all gozinto-relationships between individual combinations of (p_s, c_t) are in the source BOM represented by a single gozinto-relationship between P and C In that case some information is lost, i e the information on which individual product variant c_t, is a component of which other individual (parent) product variant p_s

In the example of the cars from the previous section, a single gozinto-relationship between an item comprising more than one car and an item comprising more than one engine, no longer contains the information which car is to be equipped with which engine It merely expresses that each car from the item *CAR* will equipped with an engine from the item *ENGINE* The information which product variant *engine* is a component of which product variant *car* must be recorded in some other way, in order to fulfil the demand that the multi-level BOM of each individual product variant can be reconstructed unambiguously from the multi-level source BOM of its item

Additional information should constitute a function which identifies the component product variant given the identification of the parent product variant In other words this function should produce a valid specification for a component item, given a valid specification of a parent item We will illustrate this problem by the example from the previous section Consider the same product variants and their initial BOM (Tables 1 and 2) but now we will distinguish other source-items, namely those listed in Table 5

Table 5

Item	Set-description
AB	$\{X \varepsilon \{x_1\}$ *AND* $Y \varepsilon \{y_1, y_2\}\}$
RQ	$\{I \varepsilon \{456, 789\}\}$
P	$\{I \varepsilon \{123\}\}$

Figure 4 depicts the source BOM which follows from the way in which items have been distinguished now Of course each gozinto-relationship has a number of attribute values, such as *quantity/per*, but they are not shown here. It will be assumed that all attributes of the gozinto-relationships have a value and that the attribute *quantity/per* has the value 1

To reconstruct the individual BOM of a product variant b from AB, the gozinto-relationships in which AB is a parent item must be retrieved to find its component items Each of these component items specifies one or more *candidate* product variants which may be the component in the BOM of b AB has two component items, namely P and RQ P is a specific item and thus consists of one product variant In this case no further specification is required. the BOM of each product variant which is a member of AB will consist of the single product variant which is a member of P

Determining the required product variant of component item RQ is more complicated Obviously the gozinto-relationship between AB and RQ does not provide enough information to determine unambiguously whether the product variant identified by $\{(I, 456)\}$ or the product variant identified by $\{(I, 789)\}$ is to become a component in the BOM of b As we have already mentioned, for this purpose additional information must be available in the gozinto-relationships of the source BOM This information should constitute a function which produces the required component product variant given a parent product variant In other words, if a parent product variant has been identified by a specification for its item, then this function should

Figure 4 The bill-of-material of item *AB*

guarantee that one specification is available for the component item, thereby identifying one component product variant This function will be called the *conversion function*

The conversion function is constituted by so called *conversion rules*. Conversion rules define deterministic relationships between parameter values of product specifications of parent product variants and component product variants In order to guarantee a *function* (in the mathematical sense), conversion rules must be formulated in such a way that, for each valid specification of the parent item, precisely one valid specification for the component item is produced Given a parent item P and a component item C and the conversion function F_{PC}, the set-descriptions D_P of P and D_C of C should be defined in such a way that for any specification S_P which is valid against P, F_{PC} generates a specification S_C for C which is a valid against C

It will be assumed that a conversion rule has the following general form.

$$(p_i, v_{ij})_{\text{component}} \text{ IF } (p_x, v_{xy})_{\text{parent}} \text{ AND } \dots \text{ AND } (p_m, v_{mn})_{\text{parent}}$$

If there are more sets of parameter values of the parent item which require the same parameter value for the component item, more than one conversion rule can be defined for that component parameter value

The conversion rules which define the relationships between parameter values of parent item *AB* and component item *RQ* are listed in Table 6 They express the fact that if a result BOM is generated for a product variant of *AB*, the product specification of the required product variant of *RQ* contains the parameter value $(I, 456)$ if the product specification from the product variant of *AB* contains the parameter value (Y, y_1) Similarly, the product variant from *RQ* contains the parameter value $(I, 789)$ in the case the product specification of the parent product variant contains the parameter value (Y, y_2)

Table 6

Parent item	seq	Component item	Conversion rules
AB	10	*RQ*	$(I, 456)$ IF (Y, y_1) $(I, 789)$ IF (Y, y_2)

Summarized, the source BOM including the conversion rules should contain all information required to generate result BOMs for individual product variants The volume of information which must be represented by conversion rules is strongly dependent on the way the source BOM has been defined and thus on the way items have been distinguished

Conversion rules will be related to a combination of a parent item and component item in the source BOM and not to the single parent item or single component item. Different parent items may define different sets of product specifications leading to the same parameter values for the component items Consider for example a second parent item for *RQ*. *FT* (see Figure 5). This parent item may consist of product variants such that, if their product specification contains the parameter value (X, x_1), the component product variant from *RQ* must be one whose product specification has a parameter value $(I, 789)$ This would require the conversion rule '$(I, 789)$ IF (X, x_1)' for the relationship *RQ gozinto FT* Obviously, the two conversion rules '$(I, 789)$ IF (Y, y_2)' and '$(I, 789)$ IF (X, x_1)', must be kept separated for the different parent items (*AB* and *FT*). The second conversion rule does not apply to the gozinto-relationship *RQ* gozinto *AB* In the generic BOM concept described in this paper it is assumed that conversion rules are related to a gozinto-relationship between a parent item and a component item This design choice may result in source BOMs with redundant conversion rules

Finally, we have implied that the parameter values of parent items and component items are different However, sometimes the parameter values applied for an item C are also applied for its parent item P The conversion rules are in this case trivial: a parameter value for the component item C is made part of the specification of C during the BOM generation process if that parameter value is also part of the specification for the parent item P Also in this case the same constraints which are a part of the set-description of the component item will also be recognized in the set-description of the parent item Recall that in this particular case each specification which

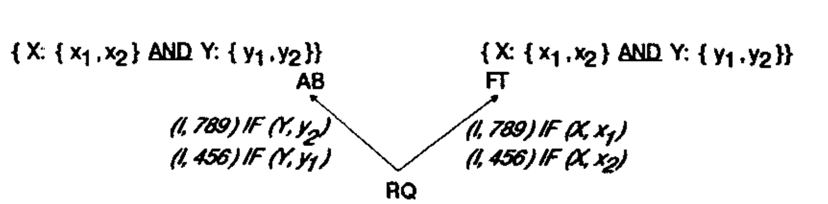

Figure 5 Conversion rules belong uniquely to one gozinto-relationship

is invalid against the component item must also be invalid against the parent item. A constraint which causes a specification to be invalid against the component item must also be defined for the parent item.

In the previous sections we have introduced the core of the generic BOM concept. The focus was on modelling (multi-level) BOMs of product variants by a (multi-level) source BOM. An important process is of course the generation of the individual BOMs of the product variants using a source BOM. Appendix 1 contains an extensive example of this process.

5. Conclusions

Since for many manufacturers product diversity will only tend to grow, generative BOM processing systems will become increasingly important. In our paper 'Generative bill of material processing systems' (Van Veen and Wortmann 1992) we show that the variant BOM concept provides a relatively simple solution, due to the choice to allow items with separate set-descriptions only to be defined at top-level in the product structure (the so-called product family items). We also show that this design choice implies a number of drawbacks.

In the generic BOM concept we have left this design choice and have taken as a starting point that an item with a separate set-description can be defined at any level in the product structure. This would avoid the drawbacks related to the variant BOM concept. However as a consequence, the concept gozinto-relationship needed to be redefined and a new concept needed to be introduced, i.e. the conversion function.

The generic BOM concept is a step forward in the developments in the area of BOM processing. Based on the core generic BOM concept as described in this paper, several additional features have been developed to support engineer-to-order and hybrid production control environments of the kind make-to-stock and assemble-to-order. For a more in-depth discussion of the generic BOM concept including these features we refer to Van Veen (1992). For more insight into BOM processing systems which are based on a generic BOM concept we refer to Hegge (1992).

References

HEGGE, H. M. H., 1992, A generic bill-of-material processor using indirect identification of product. *Production Planning and Control*, **3**, 336–342.

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, Generative bill of material processing systems. *Production Planning and Control*, **3**, 314–326.

Appendix 1. Generating a multi-level result BOM and recording it separately

It is assumed that part of the multi-level result BOM will be recorded separately for a generation identification number or for a (customer) order. One of the questions in developing a generation process for result BOMs is whether an order-dependent result BOM must comprise the complete multi-level BOM of an item down to the lowest level of the multi-level source BOM or not. In the generation process which will be described here, it will be assumed that the generation of an order-dependent result BOM stops in each path of the source BOM at the level at which a specific item is encountered, or an item is encountered which does not have a lower-level BOM. The full multi-level result BOM is obtained by exploding one or more order-dependent result BOM levels and subsequently any lower-level source BOM levels (if present).

The basic generation process for a result BOM which is recorded separately is outlined by means of a pseudo-program in Table 7. Figure 6 shows a multi-level source BOM. Tables 8 and 9 show the result of the generation process for the product variant within *ABC* which is identified by the specification $\{(X,x_1), (Y,y_2)\}$. The structure of the order-dependent result BOM is depicted in Figure 7.

It will be assumed that items in the source BOM are identified by a single item number. However, the item

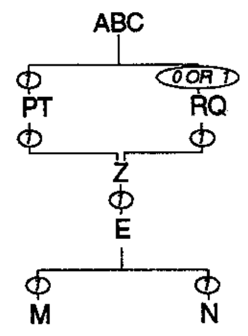

Figure 6. The multi-level source bill-of-material for generic source-item *ABC*

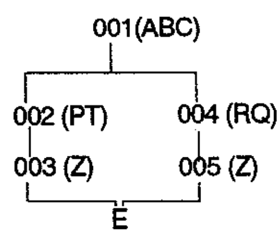

Figure 7. The multi-level result bill-of-material

Table 7

BOM explosion of item P with specification S_p recording an order dependent multi-level result BOM

```
generate result-item for P, S_P. P_result
procedure generatebomof (P, S_p, P_result)
begin while P has any more component items
        do find next relationship with parent item P. P-1.
           determine specification of P-1: S_P-i
           if qty/per of S_P- < > 0
            then get component item of P-1. C
                 if C is not a specific item
                  then determine specification of C: S_c
                       generate result - item for C with
                       S_C. C_result
                       generate result - relationship for
                       P_result, C_result, S_P-i
                       generatebomof (C, S_C, C_result)
                 else generate source/result relationship
                       for P_result, C
                 endif
           endif
    enddo
end
```

Table 8

Node	Source-item	Specification
001	*ABC*	$\{(X, x_1), (Y, y_2)\}$
002	*PT*	$\{(I, 123)\}$
003	*Z*	$\{(F, f_1), (G, g_2)\}$
004	*RQ*	$\{(I, 789)\}$
005	*Z*	$\{(F, f_2), (G, g_1)\}$

number of a generic source-item may not be distinguishing enough as a key-attribute in the order-dependent result BOM If a generic source-item *Z* is applied in more than one relationship, then it may occur that when the result BOM of an end-item is generated, more than one *different* specific item are generated from *Z*. Consider for example the item *Z* of Figure 6. In the generation process *Z* is encountered twice. Once via parent item *PT* and once via parent item *RQ*. Via these paths two different specifications were generated for *Z*. In order to represent the fact that two different specific result-items have been generated, two different entities must actually be generated. A solution to this problem is to generate a new unique number for each result-item generated For a source- item which is already specific in the source BOM, no new result-item is required The lowest level relationship of the result BOM refers to a specific source-item, or to a result- item which in turn refers to a source-item without lower-level relationships.

Table 9

Relationships			
Parent	Seq	Component	Quantity/per
001	10	002	1
001	20	004	1
002	10	003	1
004	10	005	1
003	10	*E*	1
005	10	*E*	1

The example also shows the disadvantage of the solution to generate unique item numbers in the result BOM If the same result-item had been generated from one source-item, via different paths, then two unique item numbers would still have been generated for the order-dependent result BOM. The fact that these items actually represent the same product variant is not recognizable by their item number A solution to avoid this disadvantage would be to modify the generation process such that it is checked whether the result-item about to be generated already exists. The generation process should attempt to find a result-item which has been generated from the same source-item and which has the same specification One could either choose to search for the same result-item within the scope of the set of all active result-items or to limit the scope to result-items generated within one order-dependent result BOM If the result-item about to be generated already exists within the chosen scope, then the generation process should at that point generate a result relationship which refers to the already existing result-item as a component item and not generate a new result-item