

# Rule-based Management of Product Data in CIM Systems

*W. Wöß, R. Wagner*

*Institute for Applied Knowledge Processing (FAW)*

*Johannes Kepler University Linz*

*Altenberger Straße 69, A-4040 Linz, Austria*

*Tel: +43 732 2468 9589 Fax: +43 732 2468 9308*

*e-mail: wwoess@faw.uni-linz.ac.at*

## **Abstract**

Today many enterprises have already introduced CIM systems. In many cases such enterprises are characterized by heterogeneous EDP concepts because each CIM component administrates its own databases. Due to this situation, now many CIM enterprises are confronted with high data redundancies and many inconsistencies. Data exchange between several CIM components is impossible or results in complex systems of interfaces between CIM components. In addition many enterprises have to administrate a great number of different variants of a product to fulfill today's customers' demands. The maintenance of the corresponding data with conventional methods and software systems is often made to expensive or sometimes even impossible.

Since product data are administrated in most CIM components, in this paper a rule-based product data manager is introduced, which extends conventional relational database systems with rules and operations for the dynamic generation of variant parts and variant bills of material (BOMs). Moreover, it allows the exchange of BOM data between CIM components. With this approach an efficient management of variant parts and variant BOMs is provided and redundancy and inconsistency are prevented.

## **Keywords**

CIM, extended relational databases, rule-based systems, product data management, transformation of bills of material, management of variant parts.

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35390-6\\_58](https://doi.org/10.1007/978-0-387-35390-6_58)

L. M. Camarinha-Matos et al. (eds.), *Intelligent Systems for Manufacturing*

© IFIP International Federation for Information Processing 1998

## 1 INTRODUCTION

In the last years quality and price have been the most important competition factors of an enterprise. Today many enterprises are in addition confronted with comprehensive changes of customer requirements. Many sectors changed from a provider- to a buyer-market (Poensgen 1994, Thomas 1992).

To meet these new requirements, many enterprises introduced CIM systems, hoping to fulfill customer requirements by using powerful CAD systems, PPS, customer order management systems and many other important CIM components.

Normally, this led to heterogeneous EDP concepts which are characterized by autonomous CIM components, each with its own database and with only a few possibilities for interoperability and data exchange with other CIM components in the enterprise. Due to this situation, now many CIM enterprises are confronted with the following problems:

- Individual customer production results in a huge number of different variant parts of a product again resulting in a huge number of product data. This situation is best illustrated by exemplary looking at the production statistics of the enterprise ALNO, a German furniture manufacturer. ALNO produces 4.000 final products, 70.000 (sub)parts and needs 270.000 BOMs (Straub 1990).
- The maintenance of each database with conventional methods and software systems is too expensive or even impossible (Thomas 1992).
- High data redundancies and many inconsistencies between different databases are the consequences.
- Different CIM components are often characterized by different structural views on the same product data, especially of BOM data. The consequence is, that the interoperability between different CIM components is often impossible and data exchange has to be done manually.
- Planning algorithms and knowledge about BOM processing, operations planning etc. are only implicitly defined within the whole set of variant parts.

Since CIM components store and administrate different types of data but most of them administrate some kind of product data, it can be expected, that a powerful and integrated management of product data results in a significant improvement of the whole CIM system. Therefore this paper focuses on three aspects of the management of product data:

- Exchange of BOM data between different CIM components.
- Efficient management of a large number of variant parts and variant BOMs without redundancy.
- Expert-independent availability of planning algorithms and knowledge about BOM processing, operations planning etc.

In this paper a rule-based product data manager (ProMan) is introduced, which consists of two major modules. First, the generator for variants which is

responsible for the dynamic generation of variant parts and variant BOMs. Second, the transformation processor for BOM data which is responsible for the dynamic generation of specific types of BOMs according to organizational needs.

The presented approach on the one hand significantly reduces the amount of data that has to be administrated and on the other hand reduces the necessary interaction between the operations planner and the CIM software. Dependencies and planning algorithms are transformed into a rulebase which is stored within a relational database.

The remainder of the paper is organized as follows: Section 2 starts with a discussion of two main problems with the management of product data in CIM systems. In Section 3 the concept of the rule-based product data manager is introduced. Finally, Section 4 gives some conclusions and outlines ideas for future research work.

## 2 PROBLEMS WITH THE MANAGEMENT OF PRODUCT DATA IN CIM SYSTEMS

This section gives a survey of two general problem areas in the field of product data management:

- data exchange between CAx systems and
- administration of a huge number of variant parts.

Before going into detail, the most important terms used in this paper are defined: A variant part or a variant is defined as a customer specific shape, design, etc. of a final product. The customer specifies a set of variant characteristics which describe a variant part. The set of all variant characteristics of a single variant is called variant characteristics record. The variant specification is the set of customer-specified characteristics of a variant part. That means, that the variant specification is the description of a customer specific final product. The parts family is a set of similar parts in the sense of their manufacturing technology and/or construction.

Today in many enterprises the following scenario can be found out: Different organization units of an enterprise have different structural views on the same data. For example, the design engineer creates a construction BOM with the CAD system and the operations planner needs a production BOM within the PPS system. The type of a BOM which is created within the CAD system is characterized by geometrical data and functional subassemblies, whereas the production planning requires a BOM which structures the parts and subassemblies from a production-oriented point of view. This is also expressed by different semantics of an edge within the graphical representations (trees) of the two types of BOM. Within the construction BOM an edge denotes "is part of", and within the production planning BOM the edges denote the sequence in which the parts have to be ready for assembling.

To overcome this structural problems, each software system (CIM component / CAX system) administrates its own database. The drawbacks of this trend are high data redundancies and many inconsistencies between the different databases.

Existing solutions try to solve the "data exchange problem" with standardized interfaces between two corresponding CAX systems (Brändli 1994, Grabowski 1994, Rembold 1994). But an increasing number of CAX systems results in an increasing number of necessary conversion programs. A powerful and efficient data exchange between CAX systems is not possible.

Beside the lack of a powerful data exchange between CAX systems the administration of a great number of variant parts and variant BOMs is still a problem. In principal, there are two alternatives to administrate variant BOMs (Scheer 1994). A *closed administration* of variant BOMs requires pre-defined variant parts. Thus, all variant final products are pre-defined which causes many restrictions on the flexibility of the customer specific assembly of final products. Nevertheless, the administration of variants can be done without interaction of the operations planner. To overcome the conflicts between high flexibility and automatic administration of variants, *open administration* of variants seems to be a solution. The integration of similar variant parts to a parts family is the most important concept of this principle.

The approach presented in this paper adheres to the principle of open administration of variants since this concept offers more flexibility, in particular for the maintenance of a large number of variants and variant BOMs.

### 3 THE CONCEPT OF THE RULE-BASED PRODUCT DATA MANAGER

To overcome the problems discussed in Section 1 and 2 a rule-based product data manager is introduced which is divided into the following major modules:

- The main functionality of the proposed rule-based generator for variants is that it determines the variant subpart of each subposition of the BOM of the final product, based on a customer specified set of final product characteristics.
- The rule-based transformation processor for BOM data creates an individual type of BOM for each organization unit of an enterprise.

#### 3.1 The rule-based generator for variant parts and variant BOMs

In order to establish a rule-based generator for variants and variant BOMs, different variant parts of a product are integrated to a parts family and different variant BOMs are integrated to a basic BOM. The main advantage of this approach is that the variant specific characteristics are separated from the BOM and therefore the number of BOMs is dramatically reduced. The basic BOM includes static subpositions with predefined subparts as well as open, i.e., selectable

subpositions, each together with a set of selectable subparts. "Static" means that subparts do not depend on the customer specification of the final product. In contrast to static subpositions "selectable" depicts that subparts depend on the customer specification of the final product. The dependencies between the final product and the set of selectable subparts are defined by means of if-then rules. For example if the color of the final product is green then the color of a subassembly has to be green.

The generator for variants recursively scans the BOM tree from the root to the leafs using each node as input for the determination of the variant subparts of this node. The result of the whole process is a customer specified final product together with the corresponding variant BOM.

To establish the necessary rule base, three rule types are introduced: characteristic rules, position rules and combination rules.

In general different rule types can be distinguished by their execution effects which may reach over a single or over several hierarchy levels within the BOM tree and by their inputs which may be characteristics or part numbers of variant parts. In the next sections detailed descriptions of each rule type are given by means of examples.

### *Characteristic rules*

Characteristic rules allow the definition of dependencies between characteristics of variant parts within different hierarchy levels of the BOM tree. Dependencies are possible not only between characteristics but also between explicit values of characteristics.

The data structures and the administration of BOMs are both based on one-level BOMs. Therefore the definition of characteristic rules has to consider a single hierarchy level of the BOM tree only. To allow the definition of characteristic rules over more than one hierarchy level, so called transfer characteristics are introduced. For each characteristic depending on a relevant subposition a corresponding transfer characteristic is defined.

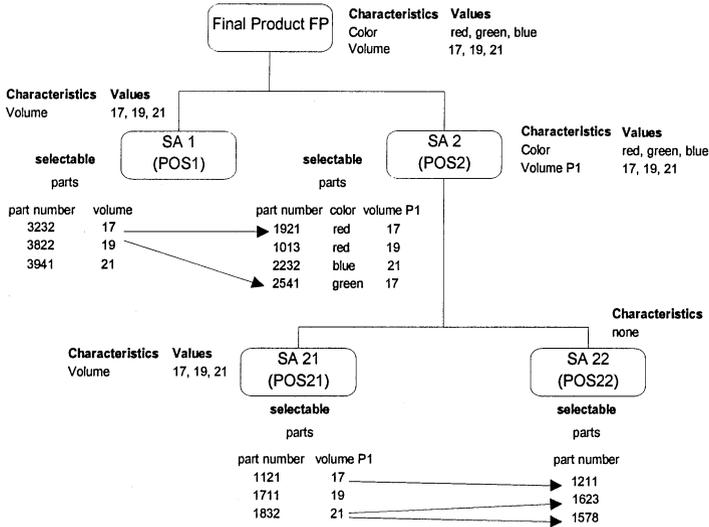
### *Position rules*

Position rules are necessary to derive the specification of a variant part of a subposition from another already specified subposition of the BOM tree. Position rules are not directly dependent on variant part characteristics and they are valid only within one hierarchy level of the BOM tree. Note, that in contrast to combination rules, rules of this type may only be specified on existing variant parts.

### *Combination rules*

Similar to position rules, combination rules are needed to specify dependencies between subpositions at the same hierarchy level of the BOM. Combination rules on the one hand are a special kind of characteristic rules, although their effect on

the other hand is similar to position rules. In contrast to position rules combination rules may be defined for variant parts that do not yet exist.



**Figure 1** Bill of material with specific rule types.

*Examples of rules*

Figure 1 illustrates a BOM containing several rules. Arrows denote the following position rules (1)-(3). Moreover, characteristic rules (4)-(5) and combination rules (6)-(7) are defined.

- |                         |                       |
|-------------------------|-----------------------|
| (1) IF POS1 3232        | (2) IF POS21 1832     |
| THEN POS2 1921          | THEN POS22 1623       |
| OR 2541                 | OR 1578               |
| (3) IF POS21 1121       |                       |
| THEN POS22 1211         |                       |
| (4) IF FP color = green | (5) IF SA2 volume P1  |
| THEN SA2 color = red    | THEN SA21 volume P1   |
| (6) IF POS1 volume      | (7) IF POS2 volume P1 |
| THEN POS2 volume P1     | THEN POS1 volume      |

If the final product FP is requested in color green, the characteristic rule (4) specifies the selectable parts 1921 and 1013 at position POS2 (SA2). Furthermore, if the volume is requested as 17, at position POS1 part number 3232 is determined. The position rule (1) defines that part number 1921 or 2541 has to be assigned to position POS2.

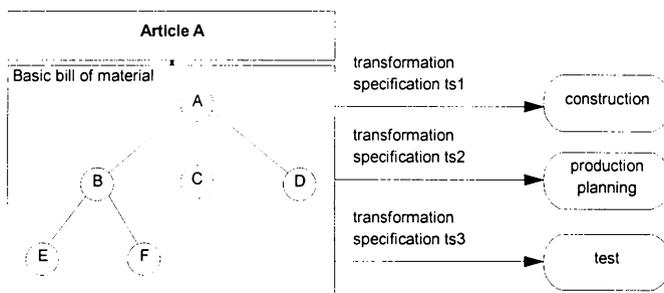
Therefore valid part numbers at position POS2 are 1921 or 2541 because of rule (1) and 1921 or 1013 because of rule (4) and color = green. Hence, the average set at position POS2 is part number 1921.

The combination rule (6) assigns 1921 or 2541 to position POS2 since the volume is set to 17. The average set of possible part numbers for position POS2 now contains part number 1921. The characteristic rule (5) assigns part number 1121 to position POS21. And the position rule (3) assigns 1211 to position POS22.

Thus, the resulting variant BOM of FP is specified as: SA1 (POS1): 3232, SA2 (POS2): 1921, SA21 (POS21): 1121, SA22 (POS22): 1211.

### 3.2 The rule-based transformation processor for BOM data

As discussed in Section 1 and Section 2, the integration of common BOM data seems to be an important step to improve the interaction of CIM components of an enterprise. But this integration process causes special problems, because it is not sufficient to integrate necessary BOM data. In addition the specific semantics of BOM data have to be considered.



**Figure 2** Basic BOM with corresponding transformation specifications.

To make the integration of BOM data of all organization units possible, this approach defines several types, e.g. a construction-, production-, order- or delivery BOM as different (structural) views of the same basic BOM. This kind of view differs in quality and complexity from the well-known view concept of relational databases, which does not support recursive algorithms and the specification of rules.

To realize such views of BOMs, a relational database system is extended with the transformation processor for BOM data. The basic BOM as well as the transformation operations for each BOM type are specified and stored within the relational database system (Figure 2). One important advantage of this approach is, that if there are updates of the BOM data necessary, it is sufficient to change the basic BOM and/or the corresponding transformation operations. Thus all specific types of BOM have a consistent state and the updates are immediately valid in all organization units.

Different types of BOMs are provided by the definition of a set of transformation specifications (set of rules) such as: "If the BOM of part A has to be transformed into type T then decompose the subtree which is located at position P". The transformation specifications are realized as a set of rules which may be defined at each node of the basic BOM, specifying those transformation operations that are necessary to transform a basic BOM into the requested type. In order to make all BOM transformations possible, five transformation operations are introduced:

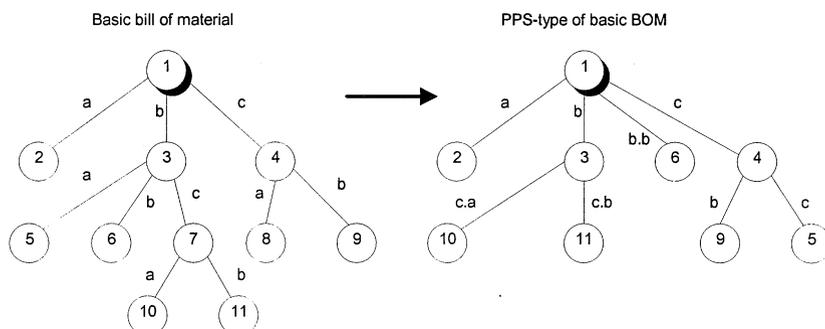
- Insert: inserts a subposition at a specified position within a BOM tree.
- Move: moves a subposition to a specified position within a BOM tree.
- High: moves a subposition one level of the BOM tree higher.
- Decompose: allows a one level decomposition of a node of a subtree. This node is substituted by its leafs.
- DecomposeAll: allows a multi level decomposition of a node of a subtree.
- Delete: removes a position including all the subpositions within a BOM tree

The following tables show the simplified relational representation of a basic BOM and corresponding transformation rules:

<i>Basic bill of material</i>			<i>Rule base</i>				
PartNo	Pos	Sub_ PartNo	BOM_Type	PartNo	S_Pos	Operation	D_Pos
1	a	2	PPS	1	b\ a	move	c\ c
1	b	3	PPS	1	b\ b	high	
1	c	4	PPS	3	c	decompose	
3	a	5	PPS	4	a	delete	
3	b	6	CAQ	1		decompose	
3	c	7				all	
4	a	8					
4	b	9					
7	a	10					
7	b	11					

In this example the first tuple of the rule base specifies that, whenever the BOM of part number 1 has to be transformed into its PPS-type, the part at part number 1, position b, subposition a (source position) has to be moved to position c, subposition c (destination position). The second tuple specifies that the part at part number 1, position b, subposition b has to be moved one hierarchy level higher. The next tuple specifies, that the subtree of part number 3, position c has to be decomposed and the last tuple specifies, that the part at part number 4, position a has to be removed. The result of the transformation process of the basic BOM into its PPS-type is shown in Figure 3.

The localization path (e.g., a, b, ...) describes a specific position. With this approach it is possible to navigate to any position within a BOM tree. For example, the localization path b\c\a (relative to part number 1) specifies part number 10.



**Figure 3** Transformation of the basic BOM of part number 1 into its PPS-type.

#### 4 CONCLUSIONS

As discussed in Section 1 and Section 2, today many enterprises are still confronted with unsolved problems in the field of product data management.

To offer a solution for the mentioned problem areas a rule-based product data manager is introduced. The implemented prototype is divided into these major modules:

- The main functionality of the proposed rule-based generator for variants is that it determines the variant subpart of each subposition of the BOM of the final product, based on a customer specified set of final product characteristics.
- The rule-based transformation processor for BOM data creates an individual type of BOM for each organization unit of an enterprise.

The approach presented in this paper on the one hand significantly reduces the amount of data that has to be administrated and on the other hand reduces the necessary interaction between the operations planner and the CIM software. Dependencies and planning algorithms are transformed into a rulebase which is stored within a relational database. Additionally each operation of the rule-based product data manager is stored within the database and therefore these operations are globally available for each CIM component. With the product data manager a system is provided that helps to bridge the gap between the business oriented and the technical software systems within a CIM factory.

Further work will be concentrated on the development of a rule-based administration of route sheets (process sheets). After that it is planned to extend the whole system with tools that help to structure existing product data within an enterprise. This will be a further step to improve the efficiency of CIM-systems.

## 5 REFERENCES

- Brändli, N., Nicolai, H., Krzepinski, A. (1994) Modellierung einer CAD/PPS-Schnittstelle, *CIM-Management*, Hrsg.: Prof. Dr. Hermann Krallmann, TU Berlin, R. Oldenbourg Verlag München, Vol. 10/1, BRD.
- Grabowski, H., Anderl R., Erb, J., Polly, A. (1994) STEP - Grundlage der Produktdatentechnologie; Aufbau und Entwicklungsmethodik *CIM-Management*, Hrsg.: Prof. Dr. Hermann Krallmann, TU Berlin, R. Oldenbourg Verlag München, Vol. 10/4, BRD.
- Poensgen, W. (1994) Prozeßorientierter Lösungsansatz für die Integration von Vertrieb und PPS bei variantenreichen Produkten *CIM-Management*, Hrsg.: Prof. Dr. Hermann Krallmann, TU Berlin, R. Oldenbourg Verlag München, Vol. 10/1, BRD.
- Rembold, U., Nnaji, B. O., Storr, A. (1994) CIM: Computeranwendung in der Produktion, Addison-Wesley, BRD.
- Scheer, A. (1994) Wirtschaftsinformatik, Referenzmodelle für industrielle Geschäftsprozesse, Springer Verlag Berlin, Heidelberg, New York, London, Paris, Tokyo, Hong Kong, Barcelona, Budapest.
- Straub, E. (1990) CIM-gerechte Organisation der Produktionsdaten: Eine moderne PPS-Lösung für Variantenfertiger ALNO, *Informationstechnik*, Hrsg.: Becker-Berke, Färber, Kroneberg, Lipp, Louis, Rogge, Schuchmann, Straßer, Zemanek, R. Oldenbourg Verlag München, Vol. 32/5, BRD.
- Thomas, R. (1992) Varianten - Herausforderung für Fertigungsunternehmen, *ZWF Zeitschrift für wirtschaftliche Fertigung und Automatisierung*, Hrsg.: Prof. Dr. Günter Spur, Carl Hanser Verlag München, Vol. 87/11, BRD.

## 6 BIOGRAPHIES

**Wolfram Wöß** received a M.Sc. and Ph.D. in Computer Science from the Johannes Kepler University in Linz, Austria, in 1993 and 1996, respectively. From 1993-1997 he was university assistant at the Institute for Computer Science of the University of Linz. Since 1998 he is university assistant at the Institute for Applied Knowledge Processing (FAW) at the University of Linz. His current research interests include CIM systems, data modeling and Internet/Intranet-based information systems.

**Roland Wagner** is full professor at the Johannes Kepler University of Linz, Austria, where he is director of the Institute for Applied Knowledge Processing (FAW). His research interests include information modeling, Internet/Intranet, tourist information systems and security.