

Value Clustering Method를 이용한 Modular BOM의 생성과 데이터베이스의 효율적인 구축에 관한 연구*

A Study on the Generation of Modular BOM and Efficient Database Construction using Value Clustering Method

지용구** · 김종한*** · 신기태**** · 박진우***

Young-Gu Ji** · Jong-Han Kim*** · Ki-Tae Shin**** · Jin-Woo Park***

Abstract

Modular BOMs are typically used in TWO-Level Master Production Schedule. To solve the problems of Modular BOM generation and efficient DB construction, we proposed Value Clustering Method. Based upon Where-Used matrix of products and components, VCM is the method to find out module by generating product family group value, product value, and component value. We also proposed method to find out information about Modules, algorithms to find out Modules that show Alternative Usage Pattern, and method to find out Modules used in a given product.

We also compared the DB creation method by Value Clustering Method and by conventional method. We compared the size of DB in both methods. We mathematically proved that the proposed method is doing better as the size and complexity of product family gets larger and more complicated.

1. 서론

MRP(Material Requirements Planning) 시스템은 기업의 생산 시스템 통제에 있어서, JIT(Just In Time), OPT(Optimized Production Technique)등과 함께 아주 유용한 방법으로 사용되어 왔으며, 기업의 생산성 향상과 재고관리에 있어서 많은 성과를 이루어 왔다. 그리고 MRP 시

스템은 기업의 변화하는 환경 및 치열한 기업간의 경쟁, 시장의 세계화, 소비자 기호의 다양화 등에 대처하기 위해 MRP II, CIM, ERP 등의 개념으로 확장되어 왔다. 그러나 이러한 개념들은 MRP시스템 본래의 기능에 새로운 영역들이 확장된 것일 뿐이며, 이러한 개념들의 활용은 MRP시스템 고유 기능의 원활한 운용을 기반으로 한다 [18].

* 이 논문은 1997학년도 대전대학교 학술연구비지원에 의한 것임.

** 생산기술연구원

*** 서울대학교 산업공학과

**** 대전대학교 산업공학과

MRP시스템을 운영하는 데 있어서 그 입력 자료로는 최종 제품의 생산량과 시점을 나타내는 기준생산계획, 재료자료, 그리고 BOM(Bill of Material)이 있다. BOM은 제조업체의 기본적인 Engineering Data의 생성, 구성, 유지, 전달하기 위한 제품과 부품의 구성 상태를 표현한 것이다. 이것은 MRP시스템 내에서 설계 변경 및 조정, 자재의 계획과 일정, 원자재계획, 조립명세서 등의 용도로 사용되고 있다[14, 15].

BOM은 그 정보를 사용하고 관리하는 주체에 따라 크게 Engineering BOM, Manufacturing BOM, Planning BOM으로 나누어진다. MRP시스템에서 기준 생산계획의 효과적인 수행을 위해 사용되는 Planning BOM은 그 사용되는 방법에 따라 다양한 이름으로 불리우고 있으며 그 중의 한 가지가 Modular BOM이다. Modular BOM은 여러 가지 제품들을 하나의 군으로 묶고 그 속에서 제품사양이나 사용형태가 같은 부품들을 모듈화하여 사용하는 것이다. 이 방법은 생산통제 환경이 계획생산에서 반제품 예측생산, 선택적 주문생산으로 변화함에 따라 기준생산계획을 효과적으로 수립하기 위해 도입된 이단계 기준생산계획(Two-Level Master Production Schedule)에 사용된다. 그러나 기존의 Modular BOM구성 방법에는 수요 예측과 판매의 관점에서 작성된 사양별 모듈에 근거하여 BOM이 구성되거나, 제품에 대한 자세한 지식을 지닌 사람에게 의해서만 BOM이 구성될 수 있는 문제점이 있다.

본 연구에서는 이러한 문제점을 극복하기 위해 Value Clustering이란 방법을 제안하고자 한다. Value Clustering 방법은 군 분류 기법(Group Technology)에서 사용되는 ROA(Rank Order Clustering Algorithm)를 Modular BOM에서 모듈을 찾아내기 위하여 처음으로 사용처 행렬(Where-Used-Matrix)에 적용시킨 방법이다. 이와 같이 ROA방법을 Modular BOM에 적용시킨 것은 Modular BOM에서 같은 사용형태를 보여주는 Module들이 ROA 방법에 의해 같은 그룹으로 묶어질 수 있기 때문이다. ROA에서는 기계-부품행렬(Machine Part Matrix)에서 그룹을 찾아내기 위하여 행과 열에 이진가중치(Binary Weight)를 주어 여러 단계를 거쳐 기계-부품 그룹을 찾아내지만 Value Clustering 방법에서는 본 연구의 대상인 제품군의 특성상 사용처 행렬의 열에만 이진가중치를 주어 단 1번의 단계로 모듈을 찾아낸다. ROA 방법은 행과

열에 이진가중치를 주어 행렬이 커지면 사용하기 곤란한 단점이 있으나 Value Clustering 방법에서는 열(제품)에만 이진가중치를 사용하여 ROA에서의 단점을 제거하였으며 열에만 적용시킬 수 있는 것은 본 연구의 대상인 제품군 내의 제품들의 수는 한정되어 있어서 모듈들은 사용처 행렬의 행(부품)에서만 나타나므로 제품들이 복잡도에 관계 없이 Value Clustering 방법을 적용시킬 수 있다. 또한 기존의 Modular BOM를 구축할 경우 모듈의 변경이나 새로운 제품의 추가 시 DB구조를 재구조화 하는 많은 문제점을 초래하지만 Value Clustering방법에서는 그와 같은 문제점을 최소화 할 수 있는 장점이 있으며 그 예는 5장에서 보여진다.

본 연구는 Value Clustering방법을 적용시켜 Modular BOM에서 필요한 모듈에 관한 정보, 선택 관계를 파악하기 위한 대체 사용 패턴(Alternative Usage Pattern) Module정보, 그리고 각 제품에 속하는 모듈(부품)을 추출하는 방법을 제시한다. 그리고 제안된 방법에 의해 관계형 DB에서 Modular BOM DB를 구축하고, Modular BOM에서 필요한 모듈에 관한 정보, Option관계를 파악하기 위한 대체 사용 패턴 Module정보와 각 제품에 속하는 모듈(부품)을 추출하는 방법을 제시하며, 이러한 방법을 DB에서 레코드 수의 측면에서 비교 분석하였다.

본 논문은 6장으로 구성되어 있으며, 2장에서는 기존의 관련 연구를 파악하고, 3장에서는 BOM 모듈을 생성하는 새로운 방법인 Value Clustering Method 및 대체 사용 패턴을 보이는 모듈을 찾아내는 방법을 제시하고, 4장에서는 예제를 보인다. 5장에서는 DB규모 측면에서 기존의 방법과 비교하고, 끝으로 6장에서 결론을 제시하고자 한다.

2. 연구 현황

앞서 언급한 바와 같이 MRP시스템에서 사용되고 있는 BOM은 그 사용 주체에 따라 크게 3가지로 구분할 수 있다. 본 장에서는 BOM과 본 연구에서 대상으로 삼고 있는 Planning BOM에 대한 연구 현황을 살펴보고자 한다.

2.1 BOM연구 현황

BOM에 대한 연구는 관계형 DB에서의 BOM자료에 관

한 문제, BOM자료에 대한 모델링, 그리고 DB에 BOM을 저장할 때의 자료 구조에 관한 연구 등의 3가지로 나누어질 수 있는데 이것들을 차례로 보면 다음과 같다.

① 관계형DB에서 BOM자료처리에 관한 연구

BOM자료는 기본적으로 제품 구조의 특성상 수직적인 구조를 지니고 있다. 따라서 이것을 관계형 DB에 저장할 경우, 부품의 정전개 또는 역전개를 할 경우 많은 질의어 연산(Query Operation)이 요구된다. 이러한 이유로 질의어 연산의 효율성을 증진시키기 위한 DB연산의 효율성 등에 관한 연구[4]가 있다. 또한 관계형DB에 저장된 BOM자료의 효율적인 추출을 위해 DB위에 지식 기반형 전문가 시스템을 도입하여 BOM자료의 효과적인 추출을 시도한 연구[23]가 있다. 이외에도 BOM자료를 저장할 때, 제품이 제조과정에서 가지는 특성과 제품을 바라보는 각 부서의 관점을 고려한 연구[3]가 있다.

② BOM자료의 모델링에 대한 연구

Parent/Child의 구조를 가지는 Generic BOM에 대한 연구[19, 20, 21, 22]가 있으며, 또한 객체 지향형 모델링 기법을 사용하여 모델링하는 연구[2, 17]에서는 BOM자료를 부품 객체를 정의하는 BOM객체와 부품의 특성을 정의하는 Property객체를 사용하여 BOM에 대한 OOBOM(Object-Oriented BOM)을 제안하였다.

③ BOM의 자료 구조에 관한 연구

BOM 자료는 크게 부품목록 화일과 제품구조 화일로 이루어져 있다. 이러한 두개의 화일을 이용하여 BOM 프로세서는 BOM자료를 나타내게 되는데, 두개의 화일은 각각 DB에서 포인터로 연결되어 있어, 한 제품에 대한 BOM구조를 알기 위해서는 여러번 DB를 접근하여야 한다. 따라서 MRP시스템에서 수많은 제품에 대하여 부품전개를 하여줄 경우, 많은 수의 DB접근으로 인하여 많은 시간이 소요된다. 이러한 단점을 극복하기 Ranjan[16]은 제품적 자료 구조에서 자료경로탐색과 자료처리 시간상의 단점을 극복한 암호화 기법을 제시하였다.

2.2 Planning BOM의 연구 현황

Planning BOM의 필요성, 종류, 생성 방법 등에 대해서 많은 저서에서 대략적인 방법을 제시하고 있다[5, 6, 10, 12, 14]. 그러나 기존의 연구에서 제시한 Planning BOM

생성 방법들은 상세한 제품 정보를 분석한 후에야 가능하고 많은 시간과 노력이 요구된다. 기업이 점차 다양화되어 가는 소비자의 요구에 대응하게 됨에 따라 제품의 사양이 급속하게 많아지고 제품 구조가 복잡하게 되어가고 있다. 따라서 Planning BOM의 구성 모델을 일관된 기준으로 생성하고 관리하는 것은 중요하며, 이러한 측면에서 여러 가지 연구가 진행되었는데 그 내용을 살펴보면 다음과 같다.

① 행렬BOM 접근법

Kneppelt[8]에 의해 제안된 방법으로 각 열에는 제품을 각 행에는 Sub-Assembly, Component, 원자재 등으로 이루어지는 표를 만들어 각 열의 제품 항목에 각 부품들의 사용 여부를 표시하는 방법이다. 이 방법을 사용하면 제품군에서 공통부품과 유일부품(Unique Parts)을 찾아낼 수 있으며 공통성에 의해 제품군을 만들어낼 수 있다. 하지만 이 방법은 모든 최종 제품을 표시할 경우 각 열이 지나치게 복잡해지므로, 각 행은 제품이 만들어지는 방법에 기초하지 않고 제품이 시장에서 팔리는 형태에 기초해야 한다. 따라서 이 방법은 제품의 복잡도가 상대적으로 작은 경우에만 유용하게 사용될 수 있다.

② 사용패턴 분석에 근거한 Modular BOM 접근법

Balcerak et al[1]은 Modular BOM을 생성시킬 때 제품의 사양 등을 고려하여 결정할 것이 아니라, 제품에 들어가는 부품들의 사용 형태를 분석하여 모듈을 생성하는 방법을 제안하였다. 같은 사용패턴을 갖는 부품은 동일 Module로 형성될 수 있으므로 기존의 개별 부품에 대하여 수작업으로 수행되던 방식을 컴퓨터를 이용한 방식으로 변경할 수 있게되어 처리시간을 절감할 수 있었다.

3. Value Clustering Method에 의한 모듈 생성

본 장에서는 서론에서 설명한 Value Clustering Method를 이용하여 모듈을 생성한 후, Modular BOM을 운영하는데 필요한 정보, 즉 모듈에 관한 정보, Option관계에 있는 정보, 그리고 각 제품이 가지는 모듈에 관한 정보를 효율적으로 추출하는 방법을 제시하고자 한다.

3.1 사용자 행렬 생성

단계 1 : 사용자 행렬 생성

하나의 제품군(Product Family Group, PFG)에 대하여 각 제품에 사용되는 Component로써 사용자 행렬을 만든다. 각 행에는 Component가, 각 열에는 제품이 위치한다. 각 제품들은 위치한 열의 순서에 의해 위치값(Position Value)을 할당받으며 제품군의 크기가 n이라면 위치값은 0부터 n-1까지의 정수값이 된다. 또한 행렬의 원소는 Component i가 제품 j에 사용되면 1, 아니면 0의 값을 가진다.

단계 2 : 특성값 계산

생성된 Matrix에서 제품군값 (PFG_Value)과 제품값(Product_Value), 그리고 부품값(Com_Value)을 생성한다.

- PFG_Value = $\sum_{i=0}^{n-1} 2^i = 2^n - 1$
- Product_Value i = 2^i
- Com_Value j = $\sum_{i \in P_j} 2^i$, P_j 는 Component j가 사용되는 제품의 위치값들의 집합.

[정리1] 제품군내에서 Component i, j(i≠j)를 이용하는 제품의 위치값들의 집합을 각각 P_i, P_j 라 할 때, $P_i \neq P_j$ 이면 $Com_Value_i \neq Com_Value_j$ 이다.

증명) Component i, j(i≠j) 에 대하여

$$|Com_Value_i - Com_Value_j| = |\sum_{a \in P_i} 2^a - \sum_{b \in P_j} 2^b| = \sum_{k_1 \in (P_i - P_j), k_2 \in (P_j - P_i)} |2^{k_1} - 2^{k_2}| \neq 0$$

표 1. 사용자 행렬 및 특성값

Component	Product					PFG_Value
	P ₁	P ₂	P ₃	P _n	
C ₁	1	1	1		1	Com_Value ₁
C ₂		1			1	Com_Value ₂
C ₃	1		1		1	.
.		1				.
.			1			.
.		1				.
.						.
C _m	1					Com_Value _m

위의 과정을 사용자 행렬에서 보면 표 1과 같이 된다.

단계 3 : 자료의 저장

단계 2에서 생성된 자료(제품-제품값, 제품-제품군값, 부품-부품값)를 DB에 저장한다. 관계형 DB에 저장 시에는 제품과 제품값, 부품과 부품값 형태로 각각의 테이블에 저장한다.

3.2 모듈 추출 방법

[정리 1]에 의해 부품값이 같은 Component들은 동일한 제품들에 사용된다. 그러므로 같은 부품값을 갖는 Component들은 하나의 모듈로 구성될 수 있다. 그리고 부품값이 제품군값과 같은 Component들은 제품군 내의 모든 제품들에 공통으로 사용되는 부품이다. 따라서 모든 제품에 공통으로 사용되는 Component를 Common Component로 정의한다. Common Component Set은 Common Component들로 이루어진 집합이다. 이 내용을 정리하면 다음과 같다.

- Com_Value_i = PFG_Value ⇒ Component_i ∈ Common_Component
- Com_Value_i = Com_Value_j ⇒ {Component_i, Component_j} ∈ Module_k, Module_value_k = Com_Value_i

Common Component Set은 Common Component Module로 정의되며, 이 정보들을 바탕으로 대체 사용 패턴 Module추출 알고리즘을 사용하여 Option관계에 있는 Module을 찾는다.

3.3 대체 사용 패턴(Module 추출 알고리즘

제품의 한 형상(Feature)에 대하여 Option관계에 있는 경우 Option을 이루는 부품들은 서로 상이하대[19, 20, 21, 22]. 전체 제품군 내에서 사용되는 형태가 서로 배타적으로 나타나는, 즉 대체 사용 패턴으로 나타나는 Module들은 한 형상에 대해서 Option관계의 Module로 간주될 수 있다. 대체 사용 패턴을 보여 주는 Module들을 추출하여 사용자가 최종적으로 Option관계에 있는 모듈들을 찾아낼 수 있는 과정은 보면 그림 1과 같다.

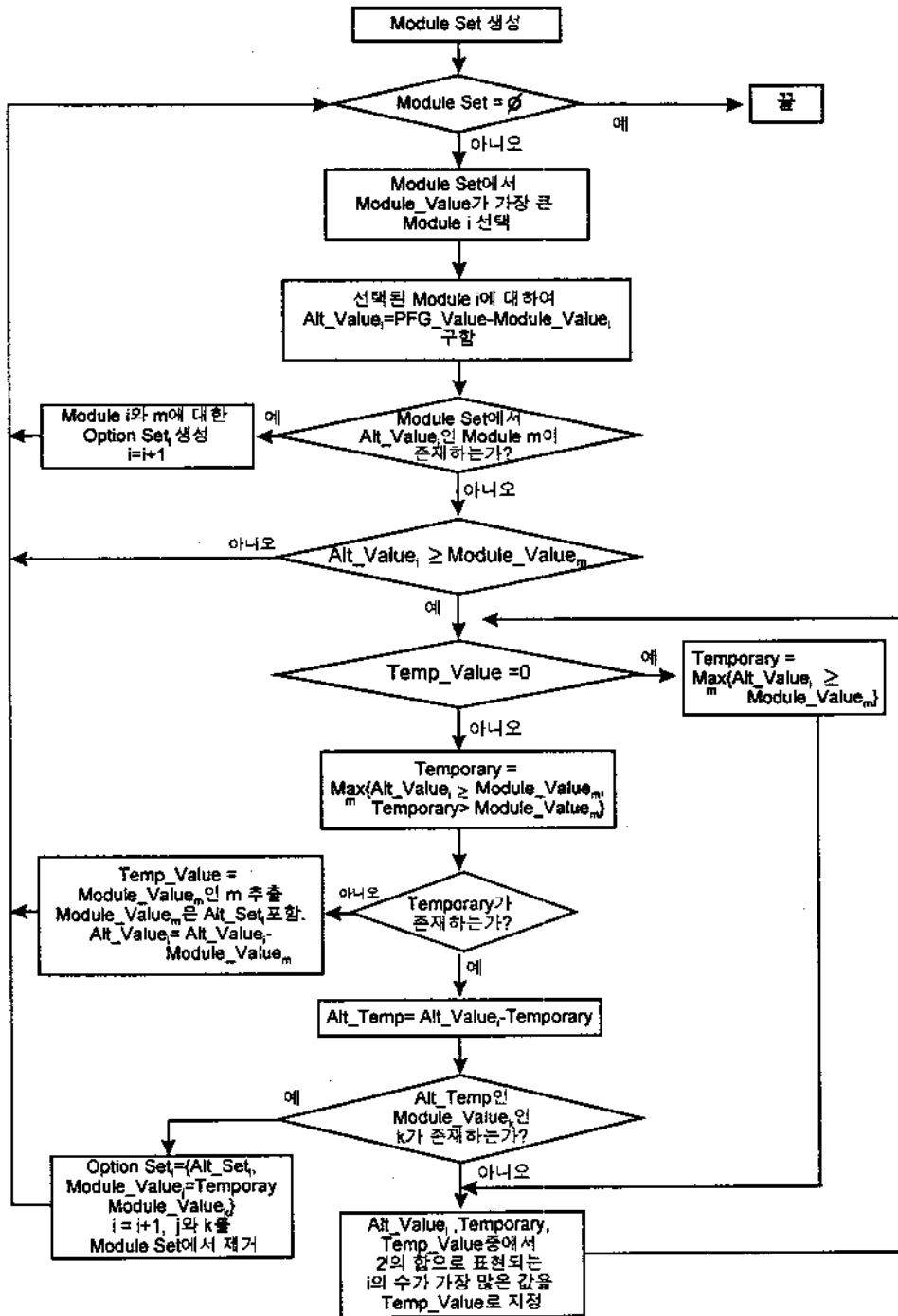


그림 1. 대체 사용 패턴 추출 알고리즘.

단계 1

- Module추출 방법을 실행하여 Module Set을 생성시킨 후 Module_Value에 의해 내림차순으로 정렬한다.
- Set $i = 1$, Set Temp_Value = 0

단계 2 : 모듈 선택

If (Module Set $\neq \emptyset$),
 Module_Value_i = Max_m{Module_Value_m},
 Module_Value_m \in Module Set,
 Alt_Value_i = PFG_Value-Module_Value_i,
 Go to 단계 3
 Else, STOP

단계 3

If $\exists j \cdot e \cdot \text{Alt_Value}_i = \text{Module_Value}_j$,
 Option Set = {Module_Value_i, Module_Value_j},
 $i \leftarrow i+1$,
 Module_Value_i, Module_Value_j를 Module Set에서 제거
 Go to 단계 2

Else

Alt_Set_i = {Module_Value_i},
 Go to 단계 4

단계 4

If $\exists m \cdot e \cdot \text{Alt_Value}_i \geq \text{Alt_Value}_m$, Go to 단계 5
 Else, Go to 단계 7

단계 5

If Temp_value = 0
 Temporary = Max_m{Alt_Value_i \geq Module_Value_m}
 Else

$$\text{Temporary} = \text{Max}_m \left\{ \begin{array}{l} \text{Alt_Value}_i \geq \text{Module_Value}_m \\ \text{Temporary} > \text{Module_Value}_m \end{array} \right\}$$

If $\neg \exists$ Temporary

Temporary = Module_Value_m과 같은 Module_Value_m을 추출
 Module_Value_m \in Alt_set,

Module_Value_m를 Module Set에서 제거,

Alt_Value_i = Alt_Value_i - Module_Value_m,
 Go to 단계 4

Else

Alt_Temp = Alt_Value_i - Temporary

If $\exists k \cdot e \cdot \text{Alt_Temp} = \text{Module_Value}_k$,

Option Set_i = {Alt_Set_i, Module_Value_i,
 Temporary, Module_Value_i},
 $i \leftarrow i+1$, Module_Value_i, Module_Value_i를 Module Set에서 제거,

Go to 단계 2

Else, Go to 단계 6

단계 6

Alt_Value_i = $\sum 2^a$ 로 표시되는 a들로 이루어진 집합 P_i,
 즉 $a \in P_i$
 Temporary = $\sum 2^b$ 로 표시되는 b들로 이루어진 집합 P_T,
 즉 $b \in P_T$ 를 생성한다.

If PZ_T \subset P_i,

If Temp_Value \neq 0

Temp_Value = $\sum 2^c$ 로 표시되는 c들로 이루어진 집합 P_{Tv},
 즉 $c \in P_{Tv}$ 를 생성한다.

If P_{Tv} \subset P_T

Temp_Value = Temporary, Go to 단계 5

Else

Go to 단계 5

Else

Temp_Value = Temporary, Go to 단계 5

Else

Go to 단계 5

단계 7

Alt_Set \subset Option Set, Go to 단계 2

3.4 제품에 속하는 모듈 추출 방법

제품의 위치값이 i인 제품 i의 제품값은 2ⁱ가 된다. 따라서 제품 i 속하는 모든 모듈(부품)들의 모듈값(부품값)에는 2ⁱ값이 포함되게 된다. 따라서 2ⁱ값을 포함하는 모든 모듈(부품)들은 제품에 사용되는 모듈(부품)들이다.

단계 1

1) Module추출 방법을 실행하여 Module Set을 생성시킨 후 Module_Value 내림차순으로 정렬한다.

2) Set $j = 1$

단계 2

1) $Module_Value_i \geq Product_Value_i$ 이면 $Module_Value_i \geq 2^k$ 를 만족하는 가장 큰 k값을 찾은 후 단계 3으로 간다.

4.1 Value Clustering 방법 적용 예

그림 2와 같은 제품군을 가지고 Value Clustering 방법에 의하여 사용처 행렬을 생성시키면 표 2와 같다.

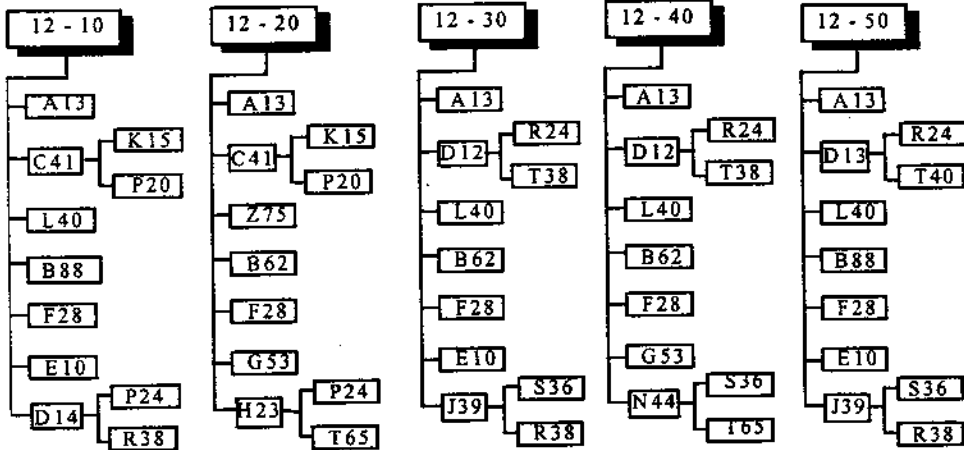


그림 2. 제품 및 구성품

2) $Module_Value_i < Product_Value_i$ 이면 단계 4로 간다.

단계 3

1) $Module_Value_i = 2^k$ 이고 $k=i$ 이면, Component j는 제품 i에 속하는 Component로 등록하고 j를 $j+1$ 로 변경한 후 단계 2로 간다.

2) $Module_Value_i - 2^k < 2^i$ 일 경우

a. $k=i$ 이면 Component j는 제품 i에 속하는 Component로 등록한 후 단계 2로 간다.

b. $k \neq i$ 이면 단계 2로 간다.

3) $Module_Value_i - 2^k > 2^i$ 이면 $Module_Value_i = Module_Value_i - 2^k$ 로 한 후 단계 2로 간다.

단계 4

제품 i에 속하는 Component로 등록된 정보를 이용한다.

표 2. 사용처 행렬

제품	12-10	12-20	12-30	12-40	12-50	제품군값
위치값	0	1	2	3	4	=31
제품값	1	2	4	8	16	부품값
A13	1	1	1	1	1	31
K15	1	1				3
P20	1	1				3
L40	1		1	1	1	29
B88	1		1		1	21
F28		1	1	1	1	31
E10	1		1		1	21
P24	1	1				3
R38	1		1		1	21
Z75						2
B62				1		10
G53				1		10
T65				1		10
R24			1	1	1	28
T38			1	1		12
S36			1	1	1	28
T40					1	16

4. 예제

앞에서 제안한 방법에 대한 예로서 Orlicky[14]에 제시된 그림 2와 같은 5개의 제품을 가진 제품군을 고려하고자 한다.

4.2 모듈 추출 방법 적용 예

Module추출 알고리즘을 적용하여 모듈을 찾아내면 다음의 표 3과 같으며 제품과 모듈과의 관계는 그림 3과 같다.

표 3. 생성된 모듈

생성된 module	Component
Common_component=31	A13, F28
Module-Value ₁ =10	B62, G53, T65
Module-Value ₂ =21	B88, R38, E10
Module-Value ₃ =3	K15, P20, P24
Module-Value ₄ =29	L40
Module-Value ₅ =28	R24, S36
Module-Value ₆ =12	T38
Module-Value ₇ =16	T40
Module-Value ₈ =2	Z75

시킨 결과는 표 3과 같다.

- ② 단계 2를 적용시켜 모듈1을 선택한다. 선택한 모듈에 단계 3을 적용시키면 다음과 같다.
Module_Value₁=29, Alt_Value₁=2, Option Set₁ = {29-[L40], 2-[Z75]}
- ③ 단계 2를 적용시켜 모듈2을 선택한다. 선택한 모듈에 단계 3을 적용시키면 다음과 같다.
Module_Value₂=28, Alt_Value₂=3, Option Set₂ = {28-[R24, S36], 3-[K15, P20, P24]}
- ④ 단계 2를 적용시켜 모듈3을 선택한다. 선택한 모듈에 단계 3을 적용시키면 다음과 같다.
Module_Value₃=21, Alt_Value₃=10, Option Set₃ = {21-[B88, R38, E10], 10-[B62, G53, T65]}
- ⑤ 단계 2를 적용시켜 모듈4을 선택한다. 선택한 모듈에 단계 3을 적용시키면 다음과 같으며 Alt Set에 4를 등록한 후 다음 단계로.
Module_Value₄=16, Alt_Value₄=15
- ⑥ 단계 4에서 Alt_Value₄=15보다 적은 Module_Value₄=12가 존재하므로 단계 5로 단계 5에서 Temporary=

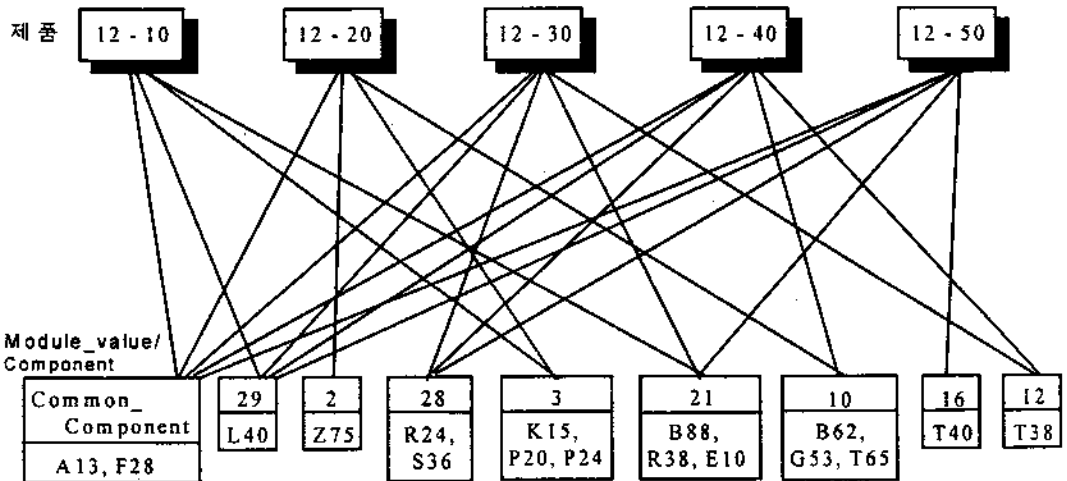


그림 3. 제품과 모듈과의 관계

4.3 대체 사용 패턴 Module 추출 알고리즘 적용 예
제품군 예제에 대체 사용 패턴 Module추출 알고리즘을 적용시키면 다음과 같은 과정을 거치게 된다.

- ① 단계 1를 적용시켜 Module Set을 생성시킨다. 생성

12이고 Alt_Temp=3과 같은 값을 지나는 Module_Value_i가 존재하지 않으므로 단계 6으로.

$$Alt_Temp = Alt_value_i - Module_Value_i$$

$$3 = 15 - 12$$

- ⑦ 단계 6에서 Module_Value₅=12=2+2³, Temp_Value=

12가 되고, 단계 5로 가서 Temporary가 존재하지 않으므로 Alt_Value₅=3을 가지고 단계 4로 감.

- ⑧ Alt_Value₅=3보다 작은 모듈이 존재하지 않으므로 단계 7로 가서 Option Set₇를 생성시킨다.

Option Set₇ = {16 - [T40], 12 - [T38]}

- ⑨ 단계 2에서 더 이상 수행할 모듈이 없으므로 과정을 끝낸다.

위에서 나온 대체 사용 패턴 Module의 결과를 보면 표 4와 같으며 나온 각 모듈들이 Option관계에 있다고 가정하고 Super BOM을 구성하면 그림 4와 같다.

표 4. 대체 사용 패턴 Module

Alternative Usage Pattern Module
Module-Value ₁ =29, Module-Value ₂ =2
Module-Value ₂ =28, Module-Value ₃ =3
Module-Value ₃ =21, Module-Value ₄ =10
Module-Value ₄ =16, Module-Value ₅ =12

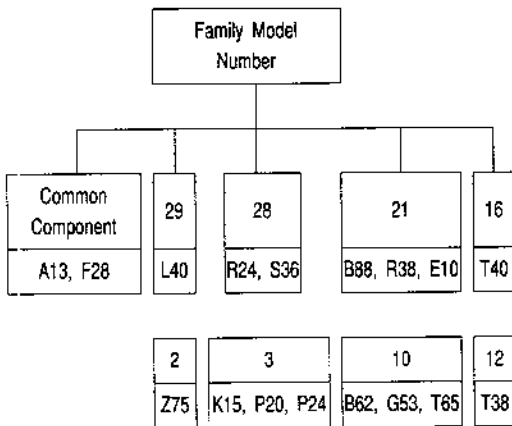


그림 4. Super BOM적용 예

4.4 제품에 속하는 모듈 추출 방법 적용 예

12-50제품에 속하는 Module을 추출한다고 하면 다음과 같은 과정을 거치게 된다.

- ① Module Set을 생성한다.

- ② 12-50의 제품값이 16이므로 16이상의 값을 지닌 모듈이 29, 28, 21, 16의 값을 지니는 모듈이다.

- ③ 단계 3을 거치면서 위의 모듈들은 모두 12-50제품에 속하는 모듈로 등록되어 사용된다.

본 연구에서 제시하지는 않았지만 하나의 Module에서 사용되는 제품에 대한 정보도 취할 수 있다. 가령 모듈값이 21인 모듈은 그 모듈값을 2의 자수승의 합으로 전개하면 (16+4+1)가 되어 이 모듈은 12-10, 12-30, 12-50에 사용되는 모듈임을 쉽게 알 수 있다.

4.5 관계형 DB에 Modular BOM정보 저장 예

본 연구에서 제안한 방법을 이용하여 [14]의 제품군 예제를 [4]의 일반적인 DB생성방법에 의한 결과와 Modular BOM으로 구성한 결과를 관계형DB에 나타내면 표 5와 같다.

표 5. 제품군 예제의 저장 예

기존의 방법	제안된 방법
{12-10, C}{12-20, C}{12-30, C} {12-40, C}{12-50, C}	{12-10, 31}{12-20, 31}{12-30, 31} {12-40, 31}{12-50, 31}
{12-10, m1}{12-10, m4}{12-10, m5} {12-20, m2}{12-20, m4}{12-20, m6} {12-30, m1}{12-30, m3}{12-30, m5} {12-30, m8}{12-40, m1}{12-40, m3} {12-40, m6}{12-40, m8}{12-50, m1} {12-50, m3}{12-50, m5}{12-50, m7}	{12-10, 1}{12-20, 2}{12-30, 4} {12-40, 8}{12-50, 16}
{C, A13}{C, F28}{m1, L40}{m2, Z75} {m3, R24}{m3, S36}{m4, K15}{m4, P20} {m4, P24}{m5, B88}{m5, R38}{m5, E10} {m6, B62}{m6, G53}{m6, T65} {m7, T40}{m8, T38}	{A13,31}{F28,31}{L40,29}{Z75, 2} {R24,28}{S36, 28}{K15,3}{P20, 3} {P24,3}{B88,21}{R38,21}{E10,21} {B62,10}{T65,10}{G53,10}{T40,16} {T38, 12}

5. DB의 규모 분석

DB 규모의 분석을 위해 필요한 기호 정의하면 아래와 같다.

P : Set of product

C : Set of Common Component

M : Set of Module

M_p : Set of Module Sets that use product p, $p \in P$

P_{m_i} : Set of Products that use Module m_i , $m_i \in M$

관계형 DB에 자료를 저장할 경우 필요한 레코드의 수를 보면 다음식 (1), (2)과 같다.

· 기존의 방법

$$|P| + \sum_{p=1}^{|P|} |M_p| + |C| + \sum_{i=1}^{|M|} |m_i| \quad \text{식(1)}$$

· 제안된 방법

$$2|P| + |C| + \sum_{i=1}^{|M|} |m_i| \quad \text{식(2)}$$

기존의 방법과 새로운 방법과의 저장시 레코드 수의 차이는 식(3)과 같다.

$$\sum_{p=1}^{|P|} |M_p| - |P| \quad \text{식(3)}$$

제품당 평균 모듈수(a)는 $a = (\sum_{i=1}^{|P|} |M_p|) / |P|$ 와 같이 정의될 수 있으므로 식(3)은 $(a-1)|P|$ 와 같이 정리된다. 그런데 a는 제품당 평균 모듈 수 이므로 제품이 복잡할수록 값이 크다고 가정할 수 있다. 따라서 위의 식은 a을 제품의 복잡도를 나타내는 상수로 놓고 제품 군의 크기에 대한 함수로 표현될 수 있다.

$$f(|P|) = (a-1)|P| \quad \text{식(4)}$$

식(4)에서 a값과 |P|값을 변화시켜 가면서 함수 값을 보면 다음의 표 6과 그림 5와 같으며, 제품군이 크고, 제품이 복잡할수록 본 연구에서 제시하는 방법이 월등히 우수함을 알 수 있다.

기존의 방법에 비해 상대적으로 제안한 방법이 모듈의 변경 시 적은 수의 레코드의 변화가 있기 때문에, 자료의 무결성(Data Integrity)을 중요시 하는 BOM DB에서는 제안된 방법이 자료의 무결성을 유지하는데 우수하다고 말할 수 있다.

표 6. α 와 |P|의 변화에 따른 $f(|P|)$ 값의 변화

n(p)	f(n(p))				
	$\alpha=1$	$\alpha=2$	$\alpha=5$	$\alpha=10$	$\alpha=100$
5	0	5	20	45	495
10	0	10	40	90	990
50	0	50	200	450	4950
100	0	100	400	900	9900
200	0	200	800	1800	19800
300	0	300	1200	2700	29700
400	0	400	1600	3600	39600
500	0	500	2000	4500	49500

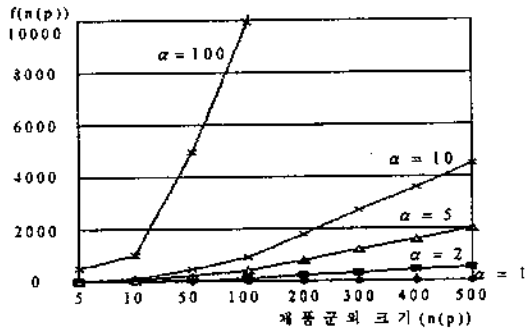


그림 5. α 와 |P|의 변화에 따른 $f(|P|)$ 값의 변화

6. 결론 및 추후 연구과제

본 논문에서는 MRP시스템에서 사용하는 Planning BOM중 Modular BOM에 대한 생성 및 효율적인 DB구축이라는 두가지 문제를 연결시켜 다루었으며 이러한 문제를 동시에 해결하기 위해 모듈 생성시 이용할 수 있는 Value Clustering 방법을 제안하였다. 그리고 Value Clustering 방법에 의해 생성된 DB에서 필요한 정보들을 도출할 수 있는 방법을 제안하였다. 그 방법들은 크게 3가지로 나눌 수 있는데 첫번째는 모듈에 관한 정보를 알아내는 방법이며, 두번째는 Option관계에 있는 Module들

을 알아내는데 필요한 정보인 대체 사용 패턴 Module을 찾아내는 알고리즘, 그리고 마지막으로는 제품이 지나는 모듈을 찾아내는 방법이다.

제안된 새로운 Value Clustering 방법에 의한 DB구축과 기존의 방법을 비교하여 DB의 규모와 DB유지의 효율성 측면에서 분석하였다. 분석결과 제품군의 크기가 커지고 제품이 복잡해 질수록 Value Clustering 방법이 기존 방법보다 우수하다는 것을 수리적으로 증명하였다.

연구 결과가 향후 현장에서 효율적으로 사용되기 위해서는 제품군에서 사용처 행렬생성시 사용되는 Component들의 수준을 기존생산계획에서 사용하는 Module에 부합되는 Component들이 되도록 사용처 행렬 생성 기준에 관한 연구와 제품이 지나는 Module을 찾아내는 알고리즘을 비트연산이나 논리연산 방식으로 변경하는 연구도 필요할 것으로 생각된다. 또한 DB의 효율성 분석에서 규모 측면뿐만 아니라 처리속도에 관한 연구도 함께 수행되어야 할 것으로 생각된다.

참고문헌

- [1] Balcerak, K. L., B. G. Dale, "Structuring modular bills of material with usage pattern analysis," *IJPR*, 1992, Vol. 30, No 2, pp. 283-298
- [2] Chung Yunkung, G. W. Fischer, "A Conceptual Structuring and Issues for an Object-Oriented Bill of Materials Data Model," *Computers Ind. Engng*, 1994, Vol 26, No.2, pp.321-339
- [3] Companys, R., P. Falster, J. L. Burbidge, *Database for Production management*, North-Holland, 1990
- [4] Date, C.J., *Database Systems Vol.I*, Addison-Wesley Publishing Company, Inc., 1990
- [5] Dave, G., *Bills of Material*, Dogwood Publishing Company Inc., 1993
- [6] Guess, V. C., *Instructor Guide Bills of Material*, APICS, 1985
- [7] Hegge, H.M.H., "A generic bill-of-material processor using indirect identification of products," *Production planning & Control*, 1992, Vol. 3, No. 3, pp. 336-342
- [8] Kneppelt, L. R., "Product Structuring Considerations for Master Production Scheduling," *Production and Inventory Management*, First Quarter, 1984, pp. 83-99
- [9] Kogan, K., "An extended bill of material model for selecting production alternatives in a multi-routeing environment," *IJ PR*, 1994, Vol. 32, No. 3, 657-667
- [10] Mather, H. F., "Design, Bills of Materials, and Forecasting - The inseparable threesome," *Production and Inventory Management*, First Quarter, 1986, pp. 90-107
- [11] Nandakumar, G., "The Design of a Bills of Material Processor Using a Relational Data Base," *Computers in Industry*, 1985, pp. 15-21
- [12] Oden H. W., et al., *Handbook of Material & Capacity Requirements Planning*, McGraw-Hill, Inc., 1993
- [13] Olsen, K. A., Saetre, P, and Thorstenson A., "A Procedure-Oriented Generic Bill of Material," *Computers and Engng*, 1997, Vol. 32, No. 1, pp. 29-45
- [14] Orlicky, J., *Material Requirements Planning*, McGraw-Hill, 1975
- [15] Plossl, K. R., *Engineering for the Control of Manufacturing*, Prentice Hall, pp. 107-126, 1987
- [16] Ranjan, B. K., "A Data Tree Structure For A Hierarchical Structure Processing," *Computer Ind. Engng*, Vol. 26, No 3, 1994, pp. 551-563
- [17] Trappey, A. J. C., Peng T., and Lin H., "An Object-Oriented Bill of Materials System for Dynamic Product Management," *Journal of Intelligent Manufacturing*, 1996, Vol. 7, pp. 365-371
- [18] Turbide, D. A., "MRPII Still Number One!," *IEE Solution*, July, 1995, pp. 28-31
- [19] Van Veen, E.A., J.C. Wortmann, "Generative bill of material processing systems," *Production Planning & Control*, 1992, Vol. 3, No. 3, pp. 314-326
- [20] Van Veen, E.A., J.C. Wortmann, "Generic bills of material in assemble-to-order manufacturing," *IJ PR*, 1987, Vol. 25, No. 11, pp. 1645-1658
- [21] Van Veen, E.A., J.C. Wortmann, "New developments in generative BOM processing systems," *Production planning & Control*, 1992, Vol. 3, No. 3, pp. 327-335

[22] Van Veen, E. A., Modelling Product Structures by Generic Bills-of-materials, Elsevier, 1992

1994, Vol. 11, No 2, pp. 65-78

[23] 박남규, 김기동, 박진우, “지식기반형 전문가 시스템을 이용한 CIM데이터베이스의 통합,” 경영과학,

97년 8월 최초 접수, 98년 4월 최종 수정