浙江科技学院学报,第 17 卷第 4 期,2005 年 12 月
Journal of Zhejiang University of Science and Technology
Vol. 17 No. 4, Dec. 2005

ERP 中 BOM 结构设计及其关键技术研究

张　玲[1],董天阳[2],曹玉华[1],张　云[1]
(1. 浙江科技学院 经济与管理学院,浙江 杭州 310023;2. 浙江工业大学 计算机软件研究所,浙江 杭州 310032)

摘　要:在设计树形结构 BOM(Bill of Material,物料清单)的基础上,着重研究了 BOM 遍历、BOM 总数量计算、BOM 快速复制等关键技术;给出了常用的两种遍历 BOM 树的算法,对其原理和算法进行了描述,并成功地将研究的 BOM 关键技术应用到实际的 ERP(Enterprise Resources Planning,企业资源计划)系统中。

关键词:物料清单;企业资源计划;物料需求计划;遍历
中图分类号:TP391　　**文献标识码**:A　　**文章编号**:1671-8798(2005)04-0268-05

Design of BOM in ERP and its related algorithms

ZHANG Ling[1], DONG Tian-yang[2], CAO Yu-hua[1], ZHANG Yun[1]
(1. School of Economics and Management, Zhejiang University of Science and Technology, Hangzhou 310023, China;
2. Institute of Software, Zhejiang University of Technology, Hangzhou 310032, China)

Abstract: Based on the design of BOM, this paper researches on the algorithms of ransacking tree BOM, the algorithms of computing the number of materials and the method of quick copy for BOM. It presents two algorithms of ransacking tree BOM and analyses the principle of them and codes. In order to validate the efficiency of these algorithms, we have succeeded in employing them to ERP system.

Key words: BOM; ERP; MRP; ransacking

BOM(Bill of Material,物料清单)是制造和装配产品所需要的物料和项目清单,它具有复杂的语义和结构特点[1]。为了将 BOM 中某种产品的所有结构部件按层次关系全部显示出来,以使产品及其构成的父子关系一目了然,一般用树形结构来描述 BOM。在数据结构中,树形结构是一类非常重要的非线性结构。直观来看,树是一种多分支多层次数据结构,由一组结点组成[2]。物料清单 BOM 列出了构成产品和部件的所有零件、部件以及它们之间的装配关系和数量关系,是生产管理的主导文件,企业的各个部门都要使用统一的物料清单进行工作。企业的许多业务都涉及到 BOM 运算,如 BOM 建立、BOM 维护、物料需求计划、制协关系调整、需求反查、生产统计等。通过对 BOM 相关业务的统计和调查,在与 BOM 相关的业务中,有些 BOM 算法是通用的,可以作为 BOM 的关键技术进行研究,在整个 ERP 系统的开发过程中都可以使用。目前,对于 BOM 的关键技术研究比较多,有 BOM 遍

收稿日期:2005-06-02
基金项目:浙江省科技中小企业技术创新基金项目(2004D70003)
作者简介:张　玲(1980—　),女,湖北公安人,硕士,助教,主要从事 ERP、工业工程、物流管理等研究。

历[3,4]、BOM 结构检查[5]等。实际上 BOM 的关键技术是比较多的，特别是单件小批量生产和面向订单装配等生产方式已成为世界制造企业公认的主要市场竞争策略，其要点是以用户多变的需求为中心，引导产品设计与制造，需要考虑产品配置和个性化产品 BOM 生成的问题。

鉴于此，本文结合开发实践，对 BOM 的结构进行设计，并对 BOM 遍历、BOM 总数量计算、BOM 快速复制等关键技术进行研究，用 Delphi 6.0 作为前台开发工具实现了这些 BOM 关键技术，并成功地应用到实际的 ERP 系统中。

1　BOM 结构设计与生成

在 Windows 平台下的 ERP 系统中，一般 BOM 采用树形结构进行构造，可以方便地实现结点的添加、删除、拖动、复制等操作，而且界面构造美观、直观易懂，用户操作简单。适应单件小批量生产方式下产品 BOM 的构造。在数据库中，笔者用 BOM 表 P_bom 和物料主文件表 P_items 一起来描述产品 BOM。其中，BOM 表对于实际生产有重要意义，它主要表达了一个产品组成信息，描述了物料的标志性属性、结构属性、输入属性和计算属性，而物料主文件表给出了各个物料的信息，对每个物料进行了详细的描述。它们之间通过物料的物料码（物料的唯一标识）联系起来描述一个物料。

在 BOM 表中，BOM 的描述属性包括：BOM 机器号（主键）、BOM 分类号、物料码、父项需要量、物料所在的层次、最低层次码、父项分类号、底层标识、图纸类型、设计购置类型等。物料清单数据库设计如表 1 所示。按照这样设计的数据库表，可以很容易生成多种形式的 BOM 格式，如：单层 BOM、多层 BOM、汇总式 BOM 等。

表 1　物料清单数据库表

序号	字段	类型	长度	空	缺省值	P	说　明
1	BOMID	INT	10	N		主键	
2	父项 ID	INT	10	N	0		
3	BOM 分类号	VARCHAR	45	N	0		每层 3 位，末层 3 位；第一层从 001 起
4	机器物料码	VARCHAR	30	N			
5	批令号	VARCHAR	30	Y			
6	层次码	INT	2	Y	0		计算生成
7	底层标识	VARCHAR	1	Y	0		计算生成
8	最低层次码	INT	2	Y	0		
9	父项分类号	VARCHAR	45	Y			
10	父项图号	VARCHAR	50	Y			
11	父项需要量	INT	8	N	0		
12	总数量	INT	8	Y	0		计算生成
13	虚拟件特征	VARCHAR	1	Y	0		虚拟组合件－1，虚拟零件－2，虚拟装配件－3
14	物料类型	VARCHAR	1	N			产品－1，部件－2，组合件－3，零件－4
15	图纸类型	VARCHAR	1	Y			自制件－1，标准件－2，外购件－3，借用件－4
16	购置类型	VARCHAR	1	Y			自制－1，采购－2
17	版本号	VARCHAR	30	Y			
18	生效日期	VARCHAR	10	Y			
19	失效日期	VARCHAR	10	Y			
24	产品代号	VARCHAR	30	N			
25	提前期	VARCHAR	1	Y			计算生成
26	建立用户	VARCHAR	8	Y			
27	…	…	…	…	…	…	其他字段

在构造 BOM 树时，BOM 树上的结点可以通过 5 个属性来描述：BOM 分类号、物料名称、物料所处的层数、父项分类号以及 BOM 机器号。在生成 BOM 树的生成中，通过定义以下的指针变量来实现：

```
type//定义结点
PLbRec = ^TLbRec; //定义指向结点的指针
TLbRec = record
code: string; //记录 BOM 分类号
fcode: string; //记录父项分类号
flag: integer; //记录物料的 Bomid 号
```

end;

2 BOM关键技术

通过对BOM相关业务的统计和调查，笔者发现部分BOM算法是通用的，在开发整个系统的过程中都可以使用。因此，应当将这些通用的算法作为BOM的关键技术进行研究。

2.1 BOM遍历

ERP系统包含一些BOM的查询模块，如展开查询、多级反查，以及工艺文件的制定，还有物料需求计划的反查等等，都需要在BOM树中检索数据。此外，为了确保MRP计算的顺利和有效进行，要进行BOM的合法性检查和计算属性生成，也需要对BOM树进行遍历，而且需要的时间往往是比较长的，因此，采用一种好的遍历方法相当必要。如何快速查到目标物料，所用到的BOM遍历算法是非常重要的。

图1是一个典型的BOM树形结构，与一般结构树的不同之处在于，在BOM表中普遍存在着同一个物料在不同层次上出现的现象，正是由于这种现象使BOM遍历处理复杂化，以致计算时间延长。从图1可以看出，物料A是物料B、C、D的父项物料，而B是E、F的父项，D是G、H、I的父项，H又是E、C的父项。产品结构BOM树之间的装配关系是一种递归关系，即一个零部件本身既可以是父件又可以是子件。同时，还应该注意到BOM结构的特殊性，即在BOM表中往往存在相同物料在不同层次的问题，另外，针对不同的计算要求往往算法也不尽相同。常用的BOM遍历算法有：前序遍历法和层次遍历法。

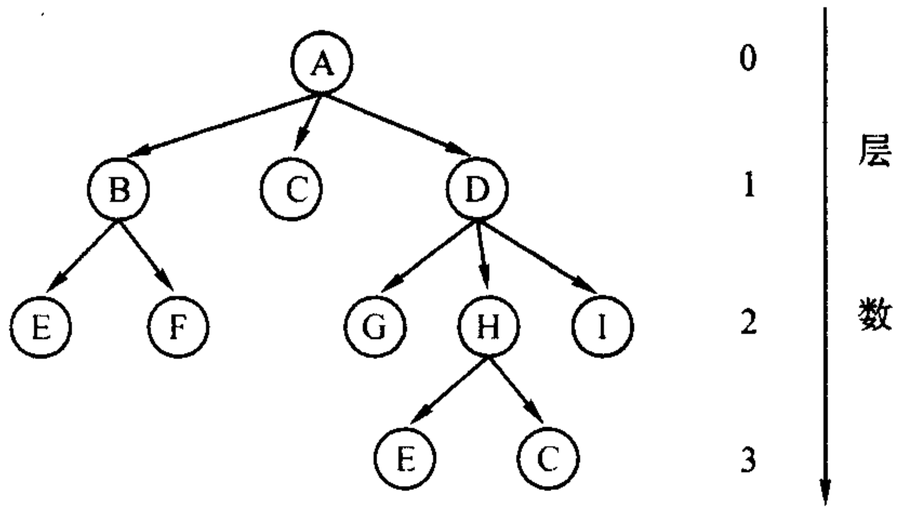

图1 BOM的树形结构

2.1.1 *前序遍历法* 在BOM树的前序遍历中，首先访问根结点的BOM分类号，然后从左到右按顺序遍历根结点的各棵子树。图1中，前序遍历的结点序列为：ABEFCDGHECI。由于在定义BOM树的前序遍历时使用的是递归算法，如果递归的层次越多，则产品BOM树的结构层次关系就越复杂。一般情况下，递归算法结构清晰，易读易理解，并且其正确性也容易验证，因此，这种遍历方法非常适宜于用递归函数实现，这样给程序的编写和调试都会带来很大的方便。在Delphi 6.0中，BOM树的前序遍历算法可以描述为：

```
procedure Re_order(Fnode: TTreenode); //
Fnode:传入根结点
begin
  if fnode.count>0 then //当前结点有子结点
    for i:=0 to fnode.count-1 do
    Re_order(fnode.item[i]);
end;
```

2.1.2 *层次遍历法* 层次遍历法首先访问处于BOM树0层上的根结点，然后从左到右依次访问处于1层上的结点，处于2层上的结点……，等等，即从上到下从左到右逐层访问BOM树各层上的结点。图1中，层次遍历的结点顺序为：ABCDEFGHIEC。在对BOM树的层次遍历法中，采用队列来存放各个结点的BOM分类号，当一层的物料访问完后，其所有的下层物料均顺序入队，如此逐层访问，则可以实现对BOM树的遍历，其流程如图2所示。

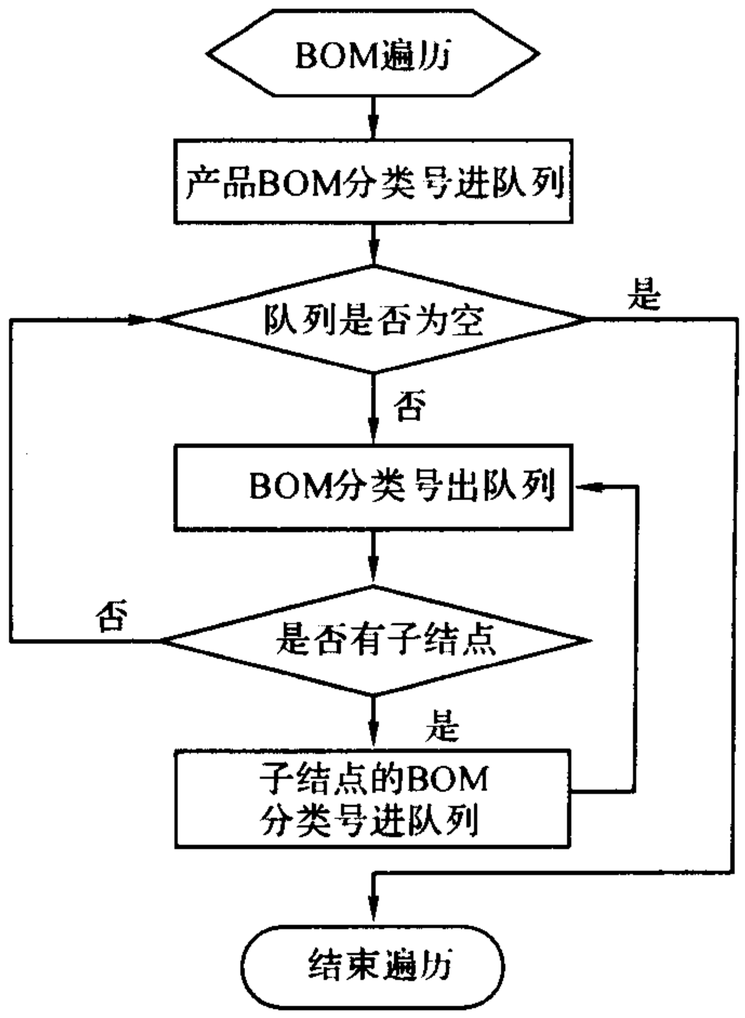

图2 层次遍历法实现BOM遍历流程图

在BOM树的分层遍历算法中引入了队列，在队列中各结点的BOM分类号以顺序方式存储。BOM树的分层遍历算法可以描述为：

```
procedure Lev_order(fnode); // fnode:传入根结点
var tpt,hpt:integer; //队列的尾标记,头标记
    Arynode:array of ttreenode;
    MaxNum, maxlevel,i:integer;
    node:ttreenode;
begin
  maxNum:=200; maxlevel:=15; //假定队列的长度为 200,树的最大层数为 15
  setlength(arynode,maxnum); //定义队列
  tpt:=0; hpt:=0;
  in_queue(fnode); //结点进队函数,若队满,返回 TRUE
  while hpt<tpt do //当头指针的值小于尾指针时,队列非空
    begin
      out_queue(node); //结点出队函数,若队空,返回 TRUE
      if node.Count>0 then
      for i:=0 to node.count-1 do in_queue(node.item[i]);
    end;
end;
```

在 BOM 树的前序遍历算法中，主要是运用函数的递归算法，该算法简洁明了，但是递归函数在运行过程中会耗费太多的时间和空间，递归算法的执行效率相对较低。而在层次遍历算法中引入的是队列，队列的一大特点就是数据的先进先出，这种算法可以充分利用数据库集合操作的特点，运行效率较高。

2.2　BOM 总数量计算

在新产品提交、更改产品提交以及 MRP 运算前的 BOM 合法性检查等功能模块中，都会对产品 BOM 的相关物料计算总数量。因此，需要将总数量计算的算法作为关键技术来研究。

总数量的计算是在最低层次码已经计算完毕的基础上进行的，计算步骤如下：

(1)定位于第一条记录(根结点)，置其总数量为 1，变量 $N=1$ 记录最低层次码。

(2) 将最低层次码 $=N$ 的相关记录的 BOM 机器号存入数组。

(3) 如果数组不空，依次数组中的 BOM 机器号：①查找相应的父项总数量。②计算本项物料的总数量：本项物料的总数量＝父项需要量×父项总数量。③如果数组的所有数据计算完毕，则 $N=N+1$，转(2)；否则，读取数组的下一个数。④如果数组为空，则总数量计算完成。

2.3　BOM 快速复制

BOM 本身比较复杂，在企业的实施 ERP 和 MRP-II 的过程中，发现企业的产品往往有很多零部件是通用的或者是类似的，这就要求 BOM 建立和维护功能支持快速拷贝功能。BOM 的快速复制与普通文字的处理是不同的，因为 BOM 的树形结构制约了 BOM 复制的实现。

笔者在 Dephi 6.0 中实现 BOM 的快速复制时，使用了一个 TTreeView 类的公共变量，用来记录 BOM 的层次关系，避免了多次循环或递归读取下属结点，以提高复制和粘贴的速度。其具体实现的关键算法如下：

```
CopyTreeRec ( fnode, fathernode: TTreenode); //fnode 源结点, fathernode 目标结点
  var i:integer; MyPlbRec:PLbRec; Mynode: TTreenode;
  begin
    New(MyPlbRec);
    MyPlbRec^. flag:= PLbRec ( fnode. Data)^. flag; //当前 bom_id
    Mynode:= PubTreeView. items. addchildobject(fathernode,fnode. Text,MyPlbRec);
    fathernode:=mynode;
    if fnode. HasChildren then
      for i:=0 to fnode. Count-1 do
          CopyTreeRec(fnode. item[i],fathernode)
    else
      exit;
    end;
```

3　BOM 管理模块的实现

在 ERP 的开发中，笔者对以上的 BOM 关键技术进行了验证。BOM 是生产管理系统中的核心数据，对其操作的基本功能单元包括：

BOM 定义：帮助用户建立产品的 BOM 表。

BOM 合法性检查：可以检查产品结构中零件下不能跟部件，标准件一定是叶子结点等。

BOM 计算属性生成：可以计算出零件总数量，

最低层次码等。

BOM 复制：为用户快捷建立类似 BOM 清单提供的功能。

图 3 是 ERP 系统中 BOM 合法性检查的运行界面，其左边的树窗口中显示的是产品 BOM。

图 3 应用程序界面

4 结 论

本文对 BOM 结构进行设计，并将 BOM 遍历、BOM 总数量计算、BOM 快速拷贝等关键技术的研究应用到实际的 ERP 中。在应用中，具体采用哪种遍历算法，要看设计的具体需求而定。一般来说，实际应用中 BOM 树的层次不会超过 15 层，如需对整个 BOM 树进行遍历，两种算法皆可选用。如果要在 BOM 树中查找一个结点，一般采用层次遍历算法。实践表明，BOM 的计算时间与产品数量有很大的关系：如果系统中只有一个产品，那么 BOM 的计算时间很快；而当系统中存在多个产品，那么即使只对一个产品进行 BOM 计算，其速度就要比原来慢得多了。BOM 作为物料需求计划展开的依据，企业的许多业务都涉及到 BOM 运算，通过对这些关键技术的研究对于提高 ERP 系统的性能具有重要意义。

参考文献：

[1] 初 壮. MRP-II 原理与应用基础[M]. 北京：清华大学出版社，1997.

[2] 严蔚敏，吴伟民. 数据结构[M]. 第二版. 北京：清华大学出版社，1992.

[3] 严志强，龚京忠，李国喜，等. 多级型 BOM 的遍历[J]. 机械设计与制造工程，2001，30(2)：33－34.

[4] 杜 杰，陆金桂. 一种改进的多级型 BOM 遍历算法[J]. 工程设计 CAD 与智能建筑，2002，(9)：65－67.

[5] 杨锦锋，方 明. 一种 ERP 物料清单结构检查方法的研究[J]. 福建电脑，2005，(1)：65，59.

(上接第 267 页)

参考文献：

[1] Breuning M M, Kriegel H P, Ng R, *et al*. LOF: Identifying density-based local outlier[R]. Dallas: In ACM SIGMOD Conference Proceedings, 2000.

[2] Fernández Pierna J A, Wahl F, Noord O E de, *et al*. Methods for outlier detection in prediction[J]. Chemometrics and Intelligent Laboratory Systems, 2002, 63(1): 27－39.

[3] Waite N B. A real-time system-adapted anomaly detector[J]. Information Sciences-Informatics and Computer Science, 1999, 115(1－4): 221－259.

[4] David Hand, Heikki Mannila, Padhraic Smyth. 数据挖掘原理[M]. 张银奎，廖 丽，宋 俊，等译. 北京：机械工业出版社，2003.

[5] Leemans V, Destain M F. A real-time grading method of apples based on features extracted from defects[J]. Journal of Food Engineering, 2004, 61(1): 83－89.