

BOM 的存储结构与遍历算法的优化及实现^①

堵士俊, 吉卫喜, 孙 斌, 钱德成

(江南大学 机械工程学院, 无锡 214122)

摘 要: 分析 BOM 的多种存储结构类型的优缺点, 给出了 BOM 实用的存储结构模型. 总结 BOM 遍历的两种基本算法: 递归算法、层次遍历算法, 在此基础上提出了产品结构树的生成策略和产品所需零件的汇总策略. 并针对一个电梯部件实例对其进行实现, 结果表明, 采用合理的存储结构及优化的 BOM 遍历算法有利于减少数据存储冗余、减少系统的开销、降低产品遍历算法的复杂性.

关键词: BOM; 存储结构; 遍历算法; 树的生成; 零件的汇总

Optimization and Realization of Storage Structure and Traversal Algorithm of BOM

DU Shi-Jun, JI Wei-Xi, SUN Bin, QIAN De-Cheng

(College of Mechanical Engineering, Jiangnan University, Wuxi 214122, China)

Abstract: The advantages and disadvantages of the polytypic BOM storage structure are analyzed. It also gives out a practical storage structure model of BOM. Two basic algorithms of traversal of BOM are summarized, which are recursive algorithm and hierarchical traversal algorithm. Based on the two basic algorithms of traversal of BOM, the generation strategy of product structure tree and the summary strategy of product parts needed are proposed. And the BOM of elevator components is realized for a specific example. The result shows that it is helpful for reducing the storage redundancy of database, system overhead and the complexity of product traversal algorithm by using reasonable storage structure and optimized BOM traversal algorithm.

Key words: BOM; storage structure; traversal algorithm; generation of tree; summary of parts

BOM(Bill of Material)物料清单, 是计算机可以识别的产品结构数据文件. 在任何制造系统中, 各部门都要从 BOM 文件中获取特定的数据, 以指导生产. BOM 详细记录了一个产品所用到的所有零部件及相关属性, 亦即父件与所有子件的从属关系、单位用量及其他属性. BOM 是 PDM/MRP/ERP 等信息化系统中最重要基础数据, 其存储结构与遍历算法设计合理与否直接影响到系统的处理性能. 因此, 根据实际的使用环境, 灵活地设计合理且有效的 BOM 是十分重要的.

关于 BOM 的存储结构, 文献[1]分析了多层式 BOM 结构和单层式 BOM 结构的优缺点后, 将两种存储结构优点结合, 得到混合式 BOM 存储结构. 文献[2]提出了一种记录结构, 用来存储向下遍历产品结构树过程中生成的信息. 文献[3]综合了单层 BOM、多层

BOM 和层次编码 BOM 三种常见 BOM 存储结构的优点, 给出了一种可配置 BOM, 能在清晰表述产品结构的同时, 减少了数据冗余.

而针对 BOM 的遍历算法, 文献[4]采用相对递归遍历的方法, 保证产品结构树的实时更新, 使之更加适应网络化的协同设计环境. 文献[5]在介绍递归查找法、分层查找法这两种算法的基础上, 提出了一种综合以上两种算法的优点的多级型 BOM 遍历算法.

1 BOM 的存储结构

1.1 常见的 BOM 存储结构

1.1.1 双亲表示法

双亲表示法中每个存储结点有两个域, 一个是数据元素域, 用以存放数据元素; 另一个是父亲指针域,

^① 收稿时间:2014-07-03;收到修改稿时间:2014-07-31

(6) 用其父项代替当前结点, 跳到第(4)步.

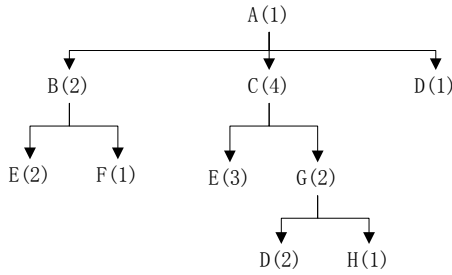


图 5 待遍历的 BOM 树

对图 5 所示的 BOM 树进行遍历的结果是 A(1)、B(2)、E(2)、F(1)、C(4)、E(3)、G(2)、D(2)、H(1)、D(1). 该算法访问 BOM 树中所有结点, 由于采用了先根遍历, 因此能够展示零部件之间的父子关系和 BOM 树的层次结构.

2.2 层次遍历算法

BOM 的层次遍历算法使用按层遍历的方法. 按层遍历采用广度优先遍历的思路, 即遍历是按 BOM 树中结点层次顺序进行, 从根结点开始, 自顶向下逐层遍历 BOM 树中各层的结点, 并对每一层的结点自左向右逐个进行遍历.

BOM 按层遍历的规则的描述是: 首先访问根结点, 再从左到右访问第二层结点, ..., 逐层访问, 直到访问完最深一层的结点. 例如, 对图 5 所示的 BOM 树按层遍历的结果是 A(1)、B(2)、C(4)、D(1)、E(2)、F(1)、E(3)、G(2)、D(2)、H(1).

该算法同样能对 BOM 树中所有结点进行访问, 并且能一次就将 BOM 树的各个零部件全部搜索出来, 提高了搜索速度, 但是由于按层遍历的特点, 遍历的结果不能展现零部件之间的父子关系, 不能用于 BOM 树的增加、删除等编辑操作.

3 BOM遍历算法的优化与应用

3.1 产品结构树的生成策略

产品结构树的生成是指系统软件通过递归算法来处理数据库中关于 BOM 的数据, 按层次和父子关系显示 BOM 的存储记录, 逐步形成一颗产品结构树的过程. 这种输出显示格式结构复杂, 而通常复杂产品的结构层次多, 组成的零部件数量也很巨大, 当涉及到递归算法时, 系统查询和显示 BOM 树的开销会过大, 只适合于产品结构树的配置和编辑处理.

基于以上情况, 而且通常用户同时只关注其中的某一段局部树结构, 没有必要一次性展开整颗产品结构树, 所以可以采取如下的产品结构树生成策略解决上述问题: 首先通过层次遍历算法生成产品结构树的第一层, 当要查看某层部件组成时, 展开该部件的 BOM 树结点, 再即时采用递归算法查询并生成其子结点. 产品结构树生成策略的流程图如图 6 所示.

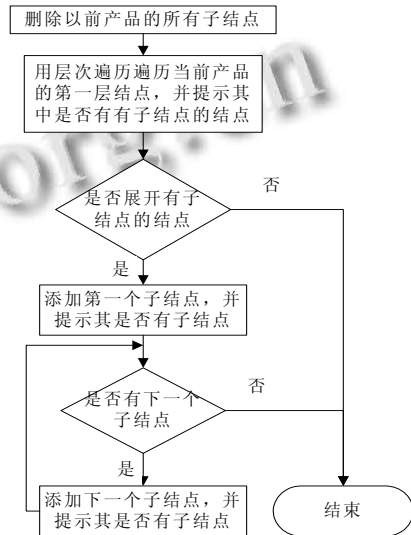


图 6 产品结构树生成策略的流程图

这样, 只要用户根据需要自行展开相应的 BOM 树结点即可. 由于每次展开一个结点只向下遍历了一层该结点的子结点, 所以实际上只进行了部分的递归算法运算. 这样对于系统的开销很低, 可以快速地显示所需要的 BOM 树结构. 而且由于是即时查询, 能够实现产品结构树的实时更新, 更加适用于现代企业的实际需要.

3.2 产品所需零件的汇总策略

产品所需零件的汇总是指对组成最终产品结构的最底层零件进行汇总, 能够反映产品所需的各种零件总数, 但不反映产品的结构关系. 如果某一零件用于产品的多个装配部件, 该汇总格式将适合确定合适的采购总量和生产总量.

基于以上所述, 可以采用优化的层次遍历算法处理数据库中的数据, 使之快速地得出产品所需零件汇总. 具体的产品所需零件的汇总策略是: 首先同样通过层次遍历算法将产品第一层中没有子结点的零件提取到汇总表内, 再将第二层中没有子结点的零件一一取出与汇总表内的零件比较, 若是相同零件, 则汇总

表内的零件的数量增加,反之,则提取到汇总表内, ..., 逐层取出并比较,直到产品的最深一层. 产品所需零件汇总策略的流程图如图 7 所示.

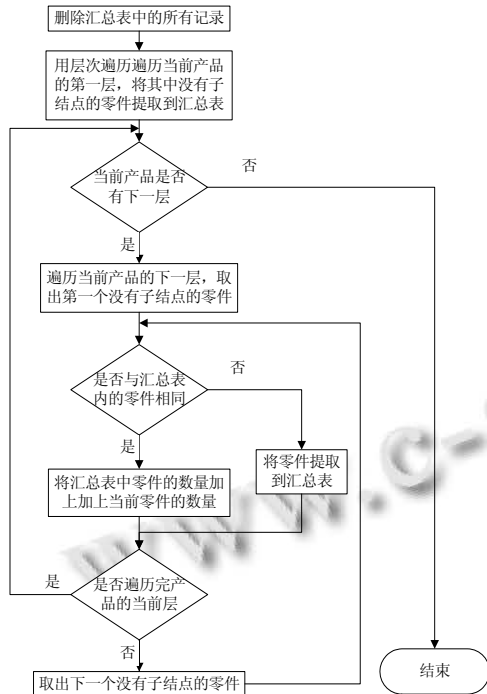


图 7 产品所需零件汇总策略的流程图

这种输出格式结构简单, 查询方便, 适合用来核算产品的成本、采购、生产等活动.

4 实例

通过对以上关键技术进行研究, 本文开发了面向电梯的 BOM 管理模块. 根据本文第 1 章 BOM 的存储结构设计, 利用面向对象的思想设计电梯 BOM 的数据库. 在 SQL Server 2008 软件中搭建完成的电梯 BOM 的数据库见图 8.

列名	数据类型	允许 Null 值
RowNo	int	<input type="checkbox"/>
BomID	int	<input type="checkbox"/>
I_jd	int	<input type="checkbox"/>
LJMC	nvarchar(100)	<input checked="" type="checkbox"/>
LJLBD	int	<input checked="" type="checkbox"/>
LJSL	int	<input checked="" type="checkbox"/>
FatherNo	int	<input checked="" type="checkbox"/>
FirstSonNo	int	<input checked="" type="checkbox"/>
NextBrotherNo	int	<input checked="" type="checkbox"/>
LevelID	int	<input checked="" type="checkbox"/>

(a) 数据库设计

RowNo	BomID	I_jd	LJMC	LJLBD	LJSL	FatherNo	FirstSonNo	NextBrotherNo	LevelID
5	1	5	制动弹簧	4	2	NULL	25	6	1
6	1	6	磁力器底座	1	1	NULL	NULL	7	1
7	1	7	整流控制器	1	1	NULL	NULL	8	1
8	1	8	松闸手柄	1	1	NULL	NULL	9	1
9	1	9	松闸指示开关	1	1	NULL	NULL	10	1
10	1	10	动心轴	1	1	NULL	NULL	11	1
11	1	11	制动螺栓锁紧螺母	1	2	NULL	NULL	12	1
12	1	12	松闸螺栓	1	2	NULL	NULL	NULL	1
13	1	13	推力杆	1	1	NULL	NULL	14	2
14	1	14	拉力杆	1	1	1	NULL	15	2
15	1	15	机壳	1	1	1	NULL	16	2
16	1	16	挡盖	1	1	1	NULL	17	2
17	1	17	纯铁盖	1	1	1	NULL	18	2
18	1	18	纯铁块	1	2	1	NULL	19	2
19	1	19	铁圈	1	2	1	NULL	NULL	2
20	1	20	臂	1	2	2	NULL	21	2
21	1	21	开口销	1	2	2	NULL	22	2
22	1	22	制动臂轴	1	2	2	NULL	23	2
23	1	23	调整垫	1	2	2	NULL	24	2
24	1	24	制动瓦	4	2	2	28	NULL	2

(b) 数据存储界面

图 8 电梯 BOM 的数据库

根据本文第 3 章 BOM 遍历算法的优化与应用, 采用面向对象的、具有可视化图形用户界面的 PowerBuilder 软件开发相应的功能模块, 实现产品结构树生成和产品所需零件汇总功能的软件界面如图 9 所示.



图 9 实现所需功能的软件界面

本文从数据库存储冗余度、查询性能等角度研究了 BOM 常用的存储结构类型的优缺点, 给出了实用的 BOM 存储结构模型, 该模型不但能查询某个产品的所有零部件组成, 能查询产生一个完整的产品结构树, 还能反查某零部件的直接或间接上级等. 本文在 BOM 遍历的两种基本算法—递归算法、层次遍历算法的基础上, 混合并优化这两种算法, 提出了产品结构树的生成策略和产品所需零件的汇总策略. 结果表明新的存储结构和优化的算法能够提高系统的效率, 减少软件对数据库系统资源的占用, 提高数据库的性能和响应能力, 实现对 BOM 的实时更新, 在实际的使用中取得了明显的效果.

参考文献

- 1 邱胜海,马银忠.先进制造环境下 BOM 的存储结构与表现形式研究与实现.现代制造工程,2008,(4):32-36.
- 2 余锐林,吴顺祥.一种改进的 BOM 展开及低层码生成算法.计算机工程与应用,2005,(27):100-102.
- 3 梁晓纯,辛国栋,刘文剑.面向游艇的可配置 BOM 生成技术.现代制造工程,2013,(2):31-36.
- 4 刘恒著,曹红松,庄德生,符美明,白友平.火炮产品结构管理模块开发的关键技术研究.现代制造工程,2012,(11):46-49.
- 5 石为人,张星,马振红,林荫华.关系型数据库 BOM 表的遍历算法的改进及实现.重庆大学学报(自然科学版),2005,28(7):82-85.
- 6 朱明方,吴及.数据结构与算法教程.北京:人民邮电出版社,2011.
- 7 薛冬娟.复杂装备制造企业物料集成管理技术研究[学位论文].大连:大连理工大学,2007.
- 8 黄改娟,张仰森,刘武雷.基于关系数据库的复合型 BOM 的设计与实现.北京信息科技大学学报,2012,27(6):66-71.
- 9 董成亮,郭顺生.一种新型的 ERP 系统中可重构 BOM 方法研究.机械制造,2008,46(522):47-50.
- 10 王辑添,司书宾.复杂装备维护 BOM 集成建模方法研究.中国制造业信息化,2011,40(23):50-54.