DOI:10.16080/j.issn1671-833x.2008.12.014

PDM 中版本管理的图模型表达方法与实现技术*

Graphic Version Model Expression Method and Implementation Technology of Version Management in PDM

西北工业大学现代设计与集成制造技术教育部重点实验室　冯向兵　莫　蓉　桂元坤

[摘要]　复杂的产品设计中会产生大量的中间版本,版本合并又是很普遍的现象,需要用图结构的版本模型进行描述。图结构模型能够直观、清晰地反映版本之间的关联关系和发展的历史记录。根据有向无环图理论,研究版本管理中的图结构模型和二维表进行图结构重构的方法,实现图结构模型的动态显示。

关键词:版本管理　图结构模型　有向无环图　产品数据管理

[ABSTRACT]　In the design of complicated products, there will be many modification versions. The combination of versions is universal, so a graphic version model(GVM) is needed to describe the combination. The GVM can represent the relation and evolution record of versions intuitionally and legibly. According to the theory of directed acyclic graph, the GVM in the version management is built, and the method of rebuilding graph for 2D table are studied and dynamical display of GVM is implemented.

Keywords: Version management　Graphic version model　Directed acyclic graph　PDM

近年来国内外在版本管理这一领域的研究较多,相当一部分涉及数据库和产品数据管理中的版本管理,还有对版本模型的研究。文献[1]按照版本数据、操作、版本历史提出了各种模型,不同的模型描述版本中的相应部分,包括数据模型、操作模型、版本模型。版本模型提供将各单个的版本集合成版本历史,利用语义支持已存在的数据;以时间戳建立面向时间的数据模型。但是在基于时间的版本中没有考虑更改的传递、信息的继承。文献[2]提出了一种新的版本模型,用来表示过程结构,把工作流看作一个复杂业务过程的核心元素,把文档和过程中的内容相互关联,形成文档版本的自动生成以及相关文档的自动配置。文献[3]建立版本关系集合模型以及版本的引用机制,对版本之间的关系连接和引用进行动态管理,能够支持产品级设计。

根据版本间的关联性,版本管理模型可以分为线性模型、树状模型和图结构模型 3 种[4-5]。在目前的版本管理系统(例如 TC-engineer 等)中,版本管理模型上存在着先天不足。这些系统大都采用的是树状的版本管理模型。树状模型无法正确地表示产品开发过程中的版本合并问题,而版本合并是企业产品开发过程中很普遍的现象。

在航空企业中,发动机产品的设计过程是一个不断反复、试探、选择、完善的复杂进化过程。以叶片设计为例,需要经过叶型设计、气动计算、叶型分析,几何设计、结构设计、强度设计等多个环节的不断修改、试验、再修改、再试验的反复迭代和试算,由此会产生大量的中间结果和多种方案。由于设计方案的差异以及不同输入参数的作用,在叶片设计中会产生同一个对象的大量中间版本。结合若干个版本的优点而继续试验获得新的中间版本,就会出现版本合并的问题。因此,对这些大量的中间版本的管理,需要一种更加有效的方法。

为了能真实、直观地反映出产品设计过程中版本间的结构关系,清晰地反映版本发展的历史记录,采用图结构模型来描述复杂的图状版本关联关系,并将这种关联关系自动、形象地显示出来,以便产品设计者对产品设计过程进行整体的把握和版本管理。

1　版本管理的模型

在产品的设计过程中,设计对象将产生一系列不同的中间版本,为了维持设计过程的正确性,不能简单地将同一对象的版本集合在一起,而是应该明显地

* 航空基金项目(05H53080)、柔性制造系统技术国家级重点实验室基金项目(51458070204HK0328)、西北工业大学研究生创业种子基金项目(Z20040017)资助。

表示出各个版本之间的关联关系。产品设计数据的版本管理有 3 种模型：线性模型、树状模型和图结构模型。

线性模型（见图 1）是最简单的一种模型，它以版本产生的先后为序，每一个版本最多只有一个父版本，且只能有一个子版本。这种模型不能区分重新设计的替换版本和修订版本的差别，不能反映不同版本之间逻辑上的关系，时间上出现在后的版本不一定是由紧接在它前面的版本得出来的，没有反映版本之间的依赖性。

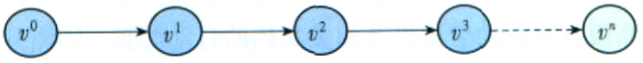

图 1　线性模型

Fig.1　Linear model

树状模型（见图 2）层次清晰，只有一个版本没有父版本，其余都有一个父版本。树状结构可以反映产品设计过程中以某一中间版本为基础选择多种设计方案从而形成多个设计结果。但是，树状结构也具有局限性，版本节点之间平行存在，版本不能有多个起始版本，不能反映多个设计版本合成一个新的版本的情况。

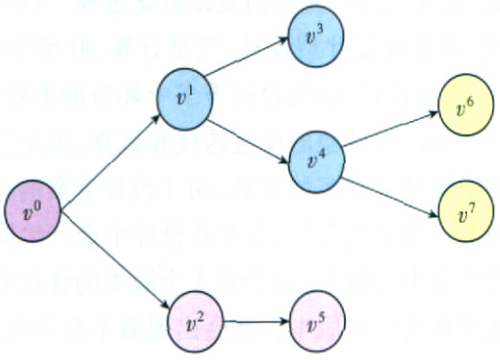

图 2　树状模型

Fig.2　Tree type model

图结构模型（见图 3）即有向无环图模型，能描述版本的历史信息，可以反映出多个设计版本融合出一个新版本的情况，是一个比较完善的版本模型。一个版本可以有多个父版本，表示版本的合并。一个版本可以有多个分支，表示版本的派生，有多个可选方案替换新版本，否则表示子节点为修订版本。图结构模型的版本既可以平行存在，又可以有多个起始版本，能真实地表示出版本之间的继承关系和依赖关系。

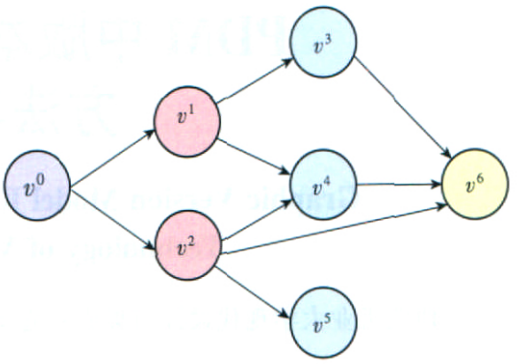

图 3　图结构模型

Fig.3　Graphic version model

由于线性模型和树状模型在结构上存在着先天的不足，不能完整地描述复杂产品大量中间版本的关联关系。随着产品设计企业的更高需求和版本管理技术的发展，图结构模型作为一种更加完善的版本模型，必将发挥更大的作用。本文以图论为基础描述叶片设计和分析中版本之间结构上的关联关系。图论便于表示版本发展中的版本派生和版本合并，能真实地描述、记录版本对象及版本之间复杂结构的关联关系。可以追溯一个版本的版本路径，体现出版本的发展过程和历史，图结构能有效描述和体现版本之间的关联关系。

2　图结构模型的建立

2.1　图结构模型定义

产品的版本图结构模型是由版本对象按照版本间的逻辑关系构成的有向无环图。图中的节点表示一个版本单元，连接 2 个版本节点之间的有向边表示版本间的继承关系。定义如下：

（1）有向图 $G=\{V, E\}$ 为一个二元组，$V=\{v^1, v^2, v^3, \cdots, v^m\}$ 为版本节点的集合。$E=\{e^1, e^2, e^3, \cdots, e^n\}$ 为连接边的集合，其中 $e^k=(v^i, v^j)$ $(k=1, 2, \cdots n; i, j=1, 2, \cdots m)$，为从版本节点 v^i 指向版本节点 v^j 的连接边；同时 v^i 为 v^j 的一个直接前驱节点，v^j 为 v^i 的一个直接后继节点，e^k 为 v^i 的一条输出连接边并同时为 v^j 的一条输入连接边。

（2）若 $V_k^a \subseteq V$，且 $V_k^a=\{v^r \mid (v^r, v^k) \in E\}(v^r \in V)$，则 V_k^a 为节点 v^k 的直接前驱版本集，记为 $\mathrm{Pre}(v^k)$；若 $V_k^b \subseteq V$，且 $V_k^b=\{v^r \mid (v^k, v^r) \in E\}(v^r \in V)$，则 V_k^b 为节点 v^k 的直接后继版本集，记为 $\mathrm{Post}(v^k)$；

航空发动机产品设计过程中产生的中间版本数

量大、关系复杂。以一个叶片在气动分析过程中产生的版本间结构关系为例，以图 3 作为此版本关系图，采用以上数学方法可以描述为：

$$V=\{v^0, v^1, v^2, v^3, v^4, v^5, v^6\};$$

$$E=\{(v^0, v^1),(v^0, v^3),(v^1, v^3),(v^1, v^4),(v^2, v^4),(v^2, v^5),(v^2, v^6),(v^3, v^6),(v^4, v^6)\};$$

当 k=4 时，$Pre(v^4)=V_4^a=\{v^1, v^2\}$；$Post(v^4)=V_4^b=\{v^6\}$。

采用版本节点表（见表 1）来记录，版本关联关系表（见表 2）来记录，就可完全记录该叶片在气动分析过程中的版本结构关系图。

表 1　版本节点表

	VersionID	VersionName	PartID
1	v^0	p^1v^0	p^1
2	v^1	p^1v^1	p^1
3	v^2	p^1v^2	p^1
4	v^3	p^1v^3	p^1
5	v^4	p^1v^4	p^1
6	v^5	p^1v^5	p^1
7	v^6	p^1v^6	p^1

表 2　版本关联关系表

	ParentID	ChindID	PartID
1	v^0	v^1	p^1
2	v^0	v^2	p^1
3	v^1	v^3	p^1
4	v^1	v^4	p^1
5	v^2	v^4	p^1
6	v^2	v^5	p^1
7	v^2	v^6	p^1
8	v^3	v^6	p^1
9	v^4	v^6	p^1

2.2　图结构模型的重构

版本节点和关联关系表在数据库中可以完全记录版本的结构图的信息。为了直观、形象地显示版本之间的发展历程，需要根据数据库中的记录重构出版本间的关联结构图，最简单的重构方法是：根据 PartID 从版本节点表中检索出一个零件（如 p^1）的所有版本节点，并分别记录这些节点，以 PartID 为条件从关联关系表中检索出该零件版本图中的边，然后按照检索的记录，每条记录对应一条边，连接相应的节点，这样重构出来的零件版本关联关系图是无误的。但是这种方法重构出来的版本图中节点位置的安排是混乱无序的，结构上也显得杂乱。

为了存储重构图结构的版本节点数据，定义链表节点类 NodeLink，结构如下：

```
Class NodeLink {
    String name;
    String[] parent;
    Point position;
    int level;
    NodeLink next; …}
```

其中 name 表示该节点在二维表中代表的 VersionID 或者唯一的标识名称 VersionName；parent 表示该节点的所有父节点的名称集，即为直接前驱版本集；position 表示该节点的显示位置；level 表示节点所处的层，如果父节点有在不同层的，那么取所在层最大的父节点的 level 加 1。

定义类 NodeList 的链表的一个空链表对象 nodeList，来存储所有的版本节点。

本课题提出一种优化的递归重构算法，将二维表存储的节点和节点之间的关联关系信息转换为适于图结构显示的数据结构。

读取图结构模型节点信息的算法如下：

（1）获取图模型的起始节点：首先从版本关联关系表中按顺序取出第一条记录的 ParentID，和版本关联关系表的所有记录的 ChildID 进行比较：a.如果有相同的情况，则 ParentID 不是起始节点，再按顺序取出下一条记录的 ParentID，与所有记录的 ChildID 进行比较；b.如果没有找到相同的情况，则此 ParentID 就为开始节点，这样的节点有且只有一个（由于版本图结构中，有且只有第一个版本号对应的版本，没有前驱版本），赋值给 node，跳出循环。

（2）采用深度优先遍历的方法 readNode（node），递归读取每个版本节点的数据：a.读取 node 节点的所有父节点，存入到字符数组 parent 中；b.计算 node 节点的所有父节点所在的最大层，再加 1 后存入到 node 节点的 level 中，即为 node 节点的层；c. 读取 node

节点的所有子节点到一临时链表 children 中，对于链表中的第 i 个子节点 child，递归调用 readNode(child)。

通过上面的读取图模型节点信息的算法，就可以将二维表中的节点和节点关联关系中的父子关系完全记录到 nodeList 的链表中，以备显示之用。

2.3 版本图元

2.3.1 版本图节点层

为了使显示的图结构模型能更贴近手绘的习惯，并适合计算机自动绘制，设计节点显示位置，使其均匀地分布到每一层上而不发生重叠，每一层高度沿起始节点的水平轴线对称设置，避免向一方偏重。首先，求得节点数最多的层的高度，层的高度指图的某一层纵向的显示的长度。然后确定节点数最多的层的高度，其他层的高度就可以根据节点数和该高度计算而得。

假设图模型有n层，定义：D_x为节点的横向显示距离常量；D_y为节点的纵向显示距离常量；N为节点的总个数；i 表示图结构的层，i=0,1,2,…,n；I 表示有最大节点数的层的集合；C_i 表示第 i 层的节点数；C_{max}表示节点数最大层的节点数；H_i 表示第 i 层的高度；H_{max}表示节点数最大层的高度；

如图 4 所示，此图有 3 层，节点最多的层为第 2 层，C_{max} 为 3，则 H_{max} 就为第 2 层的显示高度。则有：

$$C_{max}=\{\max(C_i)|i=0,1,2,\cdots,n\};$$

$$I=\{i|C_i=C_{max},\ i=0,1,2,\cdots,n\};$$

$$H_{max}=C_{max}D_y;$$

$$H_i=\begin{cases}D_y, & i=0\\ H_{max}-H_{max}(C_{max}-C_i)/C_{max}, & i\neq 0;\ i\notin I\\ H_{max}, & i\in I\end{cases}。$$

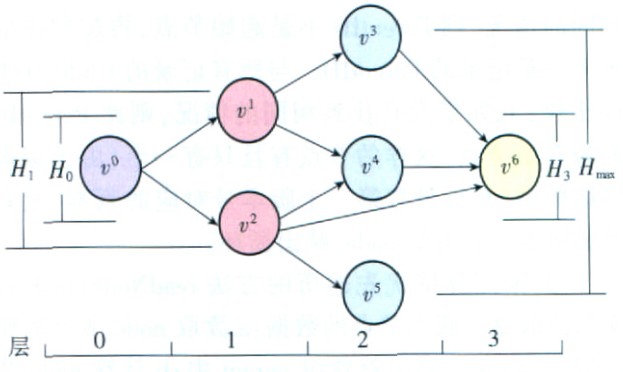

图 4 层和层的高度

Fig.4 Height among hierarechies

2.3.2 版本图节点

在确定每一层的高度以后，将节点均匀地分布到其所在层上，给出以下的定义和算法来确定每一个节点的显示位置，定义：O_x, O_y表示起始节点的绝对坐标值；L_k表示节点 v^k 所在的层，其中k=0,1,2,…,N-1；S_k表示节点 v^k 在其所在的层上的序列号，S=0,1, …C_k-1，其中k=0,1,2,…,N-1；v_x^k, v_y^k表示节点 v^k 显示的x,y坐标位置；则有：

$$v_x^k=\begin{cases}O_x, & L_k=0\\ D_xL_k+O_x, & L_k=1,2\cdots n\end{cases};$$

$$v_y^k=\begin{cases}O_y, & L_k=0\\ H_i\ (S_k/(C_i-1)-0.5)+O_y, & i=L_k=2,3,\cdots n\end{cases}。$$

将计算得到的节点v^k的坐标位置存入到链表对象nodeList的position中。根据得到的节点v^k的坐标值v_x^k, v_y^k，就可以确定版本节点的显示位置。然后计算有向边的位置，用获得两节点连线与其节点圆的交点来作为有向边的两始末端点，绘制出有向边和有向边末端的指向箭头。

3 图结构模型显示

在计算和设置得到所有需要显示的元素的各种信息(圆线位置、圆半径、线宽等)后，我们采用模型-视图-控制器(Model-View-Controller,MVC)模式来进行图结构的显示[6]。

MVC 模式能实现数据层与表示层的分离，特别适用于开发与用户图形界面有关的应用程序。在本课题的图结构显示中很好地解决了绘制的图元素的存储、更改、移动和刷新的操作，其示意图如图 5 所示。

本课题实现图结构的显示算法主要由 3 类构成，

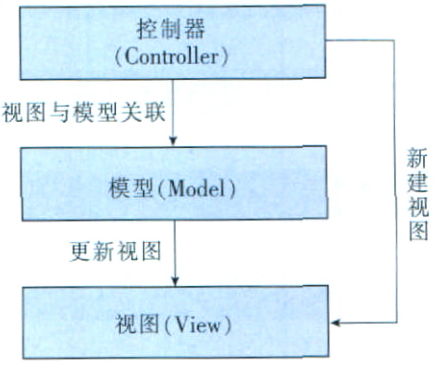

图 5 MVC 模式图

Fig.5 MVC mode

(下转第 96 页)

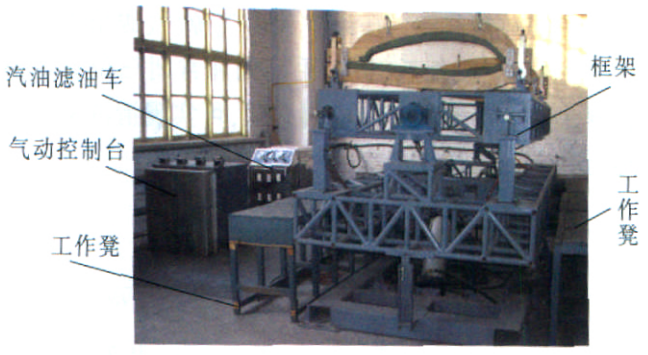

图 8 气动控制台、汽油滤油车安装位置

Fig.8 Installing position of pneumatic control table and gasoline filter vehicle

3 小鹰-500飞机机翼整体油箱清洗架安装、调试

根据试验设备安装要求及现有空间布置（见图8），气动控制台与汽油滤油车安放在一起便于操作，框架四周安装工作凳。

气动控制台和汽油滤油车所用气源压力均为0.4~0.5MPa，厂区所供压力为0.4~0.7MPa，由于管路比较长及各车间共同使用同一气源的原因，存在压力不稳的问题，通过对气动系统中气动三联件相互分离、减少气缸等措施，达到了使用要求。经过调试，气动控制台控制机翼整体油箱摇摆架运行平稳，安全可靠。

首次试车启动设备时，曾发现下层框架不能运转。经分析（参见图3）认为，下层框架以支架的中轴线和转轴为支点，通过气缸的伸缩运动实现摆动。当机翼（重50kg）被安装到上层框架后，上层框架整体的重量（由下层框架支撑）分布发生了变化。当设备启动时，由于初始支撑力较大，气缸启动时的角度形成了自锁角，使气缸的额定力矩顶不起上层框架的重量，所以气缸被锁死。后来对支座和气缸2者的支点水平距离进行了适当的调整（缩短距离），从而改善了设备载重的分布和支撑力矩状况，在不改变气缸仰角的情况下，顺利解决了问题。

4 结束语

机翼整体油箱清洗架于2006年初制造完毕，经调试及试运行完全符合设计和使用要求。机翼整体油箱安装在清洗架后，由一人负责操纵设备，即可自动运行，一个清洗过程只需26h就可完成，节约了大量的人力和物力。

（责编 依然）

（上接第92页）

分别为GraphicsModel类、GraphicsView类及GraphicsWindow类。其中GraphicsModel类扮演Model的角色，GraphicsView类为View角色，GraphicsWindow类为Controller角色。

MVC模式实现过程为：

（1）控制器GraphicsWindow类新建模型；

（2）GraphicsWindow类新建一个或多个视图对象，并将它们与模型相关联；

（3）控制器改变模型的状态，GraphicsModel类中，实现存储线、圆等各种图元数据；

（4）当模型状态改变时，GraphicsView类实现图形的绘制、更新等。

根据链表对象nodeList中的信息就能确定节点圆、有向边和指向箭头的绘制。能直观、真实地显示图3中的图结构关系图。随着节点的增加和减少，可以动态地调整节点元素的位置，完成图结构的显示。

4 结束语

本课题针对航空发动机企业复杂产品的设计过程中产生的大量复杂数据和中间版本，为了能真实、直观地反映出产品设计过程中版本间的结构关系，清晰地反映版本发展的历史记录，采用图结构模型来描述复杂的图结构版本关联关系，建立图结构模型的重构和显示方法，实现版本关系图的动态显示，以便产品设计者对产品设计过程进行整体的把握，满足航空发动机产品设计过程中版本管理的要求和复杂产品设计中的版本管理。

参 考 文 献

[1] RANDY H KATZ. Toward a unified framework for version modeling in engineering databases. ACM Computing Surveys. 1990,22(4):375~408.

[2] Raji A, Shamkant B Navathe. Version management of composite objects in CAD databases.ACM SIGMOD Record 1991,20(2):218~227.

[3] 徐立臻，徐宏炳.面向对象数据库中的版本管理.东南大学学报，1999，29(3):34~38.

[4] 张维，何卫平，刘平，等. PDM实施中的版本管理研究与应用.组合机床与自动化加工技术，2001(6).

[5] 刘方鑫，李毅. 变量化图形CAD系统中的版本管理. 计算机应用，2000，20(8):111~112.

[6] Sapia Carsten, Blaschka Markus, Hoefling Gabriele. GraMMi: using a standard repository management system to build a generic graphical modeling tool.HICSS~33 2002.（责编 钟元）