面向开发过程的产品结构形式化建模

孙 飞 唐晓青 段桂江

(北京航空航天大学 机械工程及自动化学院，北京 100191)

摘 要：为满足开发过程产品结构数据的动态结构配置、动态任务协作、动态目标求解、动态状态跟踪等应用需要，在对开发过程产品结构属性及其相互关系进行分析的基础上，提出了一种面向开发过程应用的产品结构形式化模型.结合产品开发活动的时域行为特征，给出了产品结构的时域定义，并以此为基础构建了产品结构模型的时域扩展定义.通过分析产品状态与开发任务时域行为的映射关系，在产品结构时域扩展定义的基础上，提出了基于时间截面的开发过程产品结构状态追踪方案及算法，并通过模拟产品对象在开发过程中的状态变迁过程，验证了模型及相关应用方案的实用性和有效性.

关 键 词：产品开发过程；产品结构模型；形式化建模；时间截面

中图分类号：TP 391；TH 166

文献标识码：A **文 章 编 号**：1001-5965(2008)10-1222-06

Product structure formal modeling in product development

Sun Fei Tang Xiaoqing Duan Guijiang

(School of Mechanical Engineering and Automation, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract: Based on the analysis of product structure properties and their relationships throughout product developing process, a formal product structure model was proposed to satisfy application requirements of dynamic structure configuration, dynamic tasks collaboration, dynamic objectives solution and dynamic state tracking. Considering the performance and characteristics of product developing activities in time domain, time domain definition of product structure was presented on the basis of which the product structure model extended in time domain was established. By analyzing the mapping relationship between product state and time domain performance of product developing activities, product state tracing scheme and related arithmetic based on time section were provided, which would play an important part during the process of putting the product structure model into practice. Finally, practicability and validity of the formal product structure model and its related theoretics were demonstrated by simulating a product state tracing instance.

Key words: product developing process; product structure model; formal modeling; time section

现代制造模式下，产品开发过程呈现出多企业协作、多目标求解、异地分布、动态协同的特征，产品作为开发活动的核心，产品数据必然成为连接协同开发过程中所有业务活动和动态过程的核心，成为记录开发进程的唯一载体，是产品信息的最终归属[1].产品结构是产品数据的核心，目前针对产品结构数据建模的研究比较多，但大多针对特定的应用，数据的语义一般存在于设计者和使用者的头脑中或者以一种非形式化的形式存在[1-4].对于复杂产品开发应用来说，这种局部的语义不易应用及扩展，为产品结构数据在语义层的共享与继承造成了障碍，为此，构建面向开发过程的产品结构形式化模型成为实现开发进程中信息记录与传递、业务活动运作、过程推进的关键.

收稿日期：2007-10-26

基金项目：国家高技术研究发展计划(863计划)资助项目(2003AA413040)；北京航空航天大学博士研究生创新基金资助项目

作者简介：孙 飞(1981—)，男，河南永城人，博士生，sunfei@live.com.

1　产品结构属性状态空间

状态空间是为描述事物内容、行为、特征等要素而建立起来的一种抽象参考体系,属性是产品的特征描述.属性状态空间是为描述开发过程产品结构模型内容、结构、行为、状态、特征以及开发任务执行环节各种影响要素及约束而建立的一种数据模型参考体系[5-7],如图 1 所示.

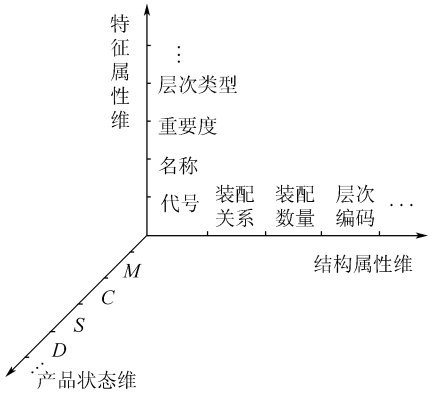

图 1　开发过程产品结构模型属性状态空间

产品结构属性状态空间由结构属性维、特征属性维、产品状态维构成(图 1),结构属性维、特征属性维分别描述产品的信息构成及结构约束,产品状态维则是产品结构数据在开发进程推进中的演化状况、数据形态的描述[8].

产品是若干部件、子部件、配件、零件等单元按照一定规则聚合在一起的零部件集合.结构属性维主要描述作为产品组成部分(组件)的零部件的构成和组合情况,结构属性维关于产品结构特征的描述是构建产品各构成部分拓扑关系的基础,产品作为一个整体系统表现出的所有行为和特征都是以各组件的结构关系为依据的.

特征属性维主要定义产品各组件的特征信息,开发过程产品组件的特征属性可以分为基本属性和扩展属性两类:基本属性是产品各组件共有特征的抽象,属于产品的固有属性,具有静态性,譬如产品名称、代号、重要度、创建时间、创建单位、创建人等;扩展属性不是所有组件的共有特征,而是具体一个(类)组件特征的抽象.开发过程中,产品组件的特征并不是完全相同的,往往会因为类型等不同而具有不同的特征信息,譬如:自加工零件需要定义材料等属性,而装配部件则不需定义材料属性;电器件需要定义电器特性,而机械件则需要定义其机械特性.这类关联于某一(类)特定组件的特征则属于扩展属性的范畴.

现代产品开发实质上是对产品进行数字化建模,并进行数据传递、拓延和加工处理的过程.产品结构定义是不断尝试、不断继承、反复迭代的,产品结构随开发进程的推进不断发展、完善[2].产品状态是指产品结构定义在开发过程各个阶段的结果及其所处的事态或情况,揭示产品结构数据的完整性及成熟度,是产品结构在开发过程中时域行为及特征的体现[3,8].工程应用中,一般会对开发过程进行阶段性划分,不同阶段产品结构的成熟度也不同,进而反映为产品状态的变迁,譬如,某产品开发过程被划分为模样(M)、初样(C)、试样(S)、定型(D)等阶段,各阶段产品结构定义的内容不同,处于不同的产品状态.产品状态维主要描述开发过程各阶段产品结构的状态,记录产品结构随开发进程推进的发展、演化轨迹,管理开发过程中各状态产品结构数据之间的继承、约束关系,为产品结构数据的管理及历史追溯提供依据.

2　开发过程产品结构属性构成

对产品结构属性构成进行探讨是进行形式化建模的前提,属性状态空间为探讨产品结构属性构成提供了相应的参考体系.开发过程产品结构属性构成如图 2 所示.

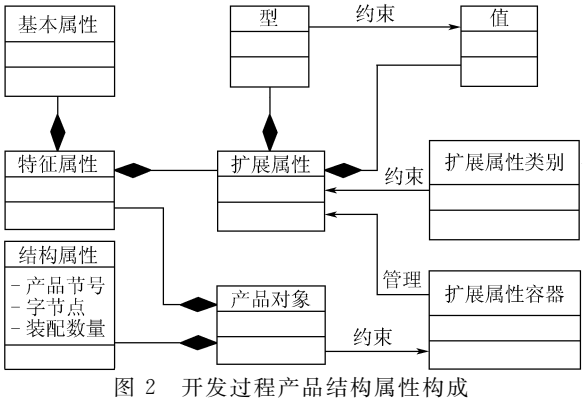

图 2　开发过程产品结构属性构成

开发过程产品结构的特征可以细分为结构属性、基本属性和扩展属性(图 2).其中结构属性以产品装配关系为基础描述开发过程产品的组件构成情况;基本属性则是开发过程中相对固定、通用的产品静态特征的描述,譬如产品在开发任务执行体系中的代号、名称等.结构属性、基本属性是产品各组件的共有特征,在表达方式上可以采用较为固定的结构来描述.

扩展属性是为应对开发过程产品特征的个性化、动态性应用而扩展的.扩展属性不是产品各组件共有属性,而是某一(类)组件的特有特征属性,

这决定了扩展属性具有“型”和“值”的概念，型是对某一扩展属性的说明，值则是扩展属性的具体赋值. 扩展属性与具体的产品组件直接关联，而不是关联于所有产品组件. 工程应用中，通过扩展属性容器对扩展属性进行统一管理，使业务人员可以根据产品各组件的个性化需要，动态定义并维护其属性，从而实现了产品组件个性化属性的描述需要，提高产品信息描述的灵活性及适应性.

3 产品结构形式化定义

开发过程产品结构不仅仅是作为产品组分的零部件的简单集合，还包括零部件之间的构成关系，即产品结构应该包括属性对象和结构对象. 以下分别给出它们的定义.

定义 1 属性对象

$$N=\{n\mid n=\langle i^n;A_i^n\rangle \wedge i^n\in I^n \wedge A_i^n\subseteq A^n\}$$

定义 1 中，属性对象 n 被定义为一系列产品特征属性的集合，其中 I^n 为属性对象的唯一标识集；$A^n=A_b^n\cup A_e^n$ 为产品的特征属性集，包括产品的基本属性和扩展属性，A_b^n 代表产品的基本属性集，A_e^n 代表产品的扩展属性集，A_i^n 为属性对象 $n|_{i^n}$ 具有的属性集合，是 A^n 的子集[9].

定义 2 结构对象(装配关系)

$$R=\{r\mid r=\langle i_1^n,i_2^n,x\rangle \wedge i_1^n\in N\wedge i_2^n\in N\wedge x\in \mathbf{Z}\}$$

$$R^+=\{r\mid r=\langle i_1^n,i_2^n,x\rangle\in R\wedge x\in \mathbf{Z}^+\}$$

$$R^-=\{r\mid r=\langle i_1^n,i_2^n,x\rangle\in R\wedge x\in \mathbf{Z}^-\}$$

其中，$R\subseteq N\times N\times \mathbf{Z}$ 是产品的装配空间，$\mathbf{Z}$ 表示整数集，$x\in\mathbf{Z}$ 表示产品组件间的装配数量，具体意义为：当 $x>0$，属性对象 i_1^n 和 i_2^n 有直接的父子关系(正装配关系)，且装配数量为 x；当 $x<0$，属性对象 i_1^n 和 i_2^n 有直接的子父关系(负装配关系，即 i_1^n 是 i_2^n 的子节点)，且装配数量为 $-x$，定义负装配关系，主要为了兼顾复杂产品开发过程中历史数据的逆向追溯应用，通过该对象可以方便地从产品子节点逆向追溯到任意层次的父节点；当 $x=0$，属性对象 i_1^n 和 i_2^n 没有直接的装配关系. R^+ 是正装配关系的集合(正装配空间)，R^- 是负装配关系的集合(负装配空间). 装配空间 R 上，有 $R^+|_{N\times N}=(R^-|_{N\times N})^{\mathrm{T}}$，即正负装配空间 R^+ 和 R^- 在 $N\times N$ 上的投影互逆，其中，$\{\langle a,b\rangle\mid a\in N,b\in N\}^{\mathrm{T}}=\{\langle b,a\rangle\mid a\in N,b\in N\}$[9].

属性对象和结构对象分别对应于产品的实体部分和构成关系，二者共同定义了产品组件及其在应用中的局部环境. 定义 1 和定义 2 的基础上，可以给出产品对象的定义.

定义 3 产品对象

二元组 $P=\{N,R\}$ 称为产品对象的充分必要条件是：

1) $N=\{n\mid n=\langle i^n;A_i\rangle \wedge i^n\in I^n\wedge A_i\subseteq A^n\}$

2) $R=\{r\mid r=\langle i_1^n,i_2^n,x\rangle \wedge i_1^n\in N\wedge i_2^n\in N\wedge x\in\mathbf{Z}\}$

其中，N 是产品属性对象；R 是产品的结构对象[4]. 定义 3 中给出的产品对象内涵突破了传统产品树的概念，把产品的树结构扩展成一个复杂有向网络，在该网络中顶点是属性对象，顶点之间的有向边表示正装配关系，有向边权值则是装配关系的装配数量，如图 3 所示.

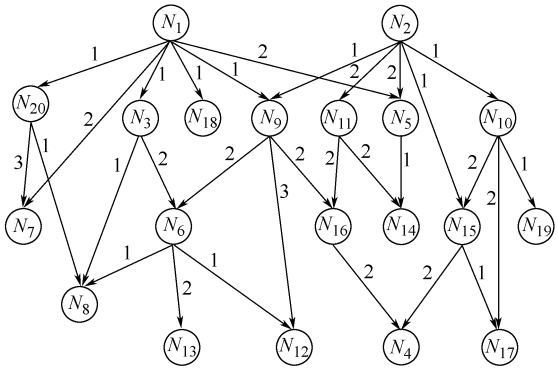

图 3 产品对象的网络结构

产品对象的复杂网络结构(图 3)是对传统产品树结构的扩展，为描述开发过程中系列产品之间结构的复用、演化提供了理论支持，实现了对开发过程产品结构外延的更全面覆盖[4,10]. 图 3 所示产品对象的复杂网络结构中，产品节点的内涵包括代表节点实体的属性对象和定义在装配空间 $N\times N\times\mathbf{Z}$ 上的局部应用环境，从这个意义上可以给出产品节点对象的定义.

定义 4 节点对象

$$\bar{r}=\overline{\langle i_1^n,i_2^n,x\rangle}=\begin{cases}\{r,n|_{i_2^n}\} & x>0\\ \varnothing & x=0\\ \{r,n|_{i_2^n}\} & x<0\end{cases}$$

产品的节点对象被定义为 $N\times N\times\mathbf{Z}$ 上属性对象和结构对象的集合，在装配空间上，节点对象 $\bar{r}=\overline{\langle i_1^n,i_2^n,x\rangle}$ 由结构对象 $r=\langle i_1^n,i_2^n,x\rangle$ 唯一确定，而属性对象可以同时隶属于多个产品节点. 可以把产品对象定义为节点对象的集合：

定义 3′ 产品对象 $P=\{\bar{r}\mid r\in R\}$

4 产品结构时域扩展定义

4.1 产品结构的时域本质

属性状态空间中，产品对象的内容随着开发

任务的执行过程逐渐发展、拓延[2,8],并通过产品状态维来描述产品结构的发展、演化特征(图 1).由于产品开发是一个时域上连续的任务执行过程,从系统角度看,开发过程产品结构与时间之间必然客观存在着某种映射关系 $P:t\rightarrow P$,该映射关系能够反映时域上连续的时刻与产品对象内容的对应关系,可以给出产品对象的时域定义:

$$P=P(t)=P_0+\int_{t_c}^{t} p(t)\cdot \mathrm{d}t \quad t\in[t_c,t_d] \tag{1}$$

其中,P 为产品对象的内容,由两部分构成:产品对象在开发过程的时域积累 $\int_{t_c}^{t_d} p\cdot \mathrm{d}t$ 以及已有内容 P_0(相似产品继承),t_c 为开发过程启动时间,t_d 为终止时间;t 为开发过程中的任意时刻;$p(t)$为产品对象内容的增量函数,由于产品对象内容随开发进程的推进逐渐丰富,故 $p(t)\geqslant 0$.

4.2　产品结构定义的时域扩展

产品对象的时域定义从根本上揭示了其时域继承性及延续性,反映了产品对象状态演化的深层本质,为产品结构数据的状态管理及应用提供了理论支持.为在执行环节实现产品对象的状态管理及应用,需要把产品结构定义的时域行为及特征映射到产品定义中去,以下给出产品对象的时域扩展定义.

定义 5　扩展属性对象

$$N'=\{n' \mid n'=\langle i^n,t^{\mathrm{cre}},t^{\mathrm{inv}};A_i^n\rangle \wedge i^n\in I^n \wedge A_i^n\subseteq A^n \wedge t^{\mathrm{cre}}\in T \wedge t^{\mathrm{inv}}\in T\}$$

定义 6　扩展结构对象(装配关系)

$$R'=\{r' \mid r'=\langle i^m,t^{\mathrm{cre}},t^{\mathrm{inv}};r\rangle \wedge i^m\in I^m \wedge r\in R \wedge t^{\mathrm{cre}}\in T \wedge t^{\mathrm{inv}}\in T\}$$

定义 5 和定义 6 分别是在属性对象和结构对象定义的基础上通过添加状态标签(生效状态标签 t^{cre}、失效状态标签 t^{inv})扩展而来,T 表示时间.状态标签是产品对象时域行为在其模型结构的映射,并最终反映为产品结构状态的变迁.

定义 7　扩展产品对象

二元组 $P'=\{N',R'\}$称为产品对象的充分必要条件是:

1) $N'=\{n' \mid n'=\langle i^n,t^{\mathrm{cre}},t^{\mathrm{inv}};A_i\rangle \wedge i^n\in I^n \wedge A_i\subseteq A^n \wedge t^{\mathrm{cre}}\in T\wedge t^{\mathrm{inv}}\in T\}$;

2) $R'=\{r' \mid r'=\langle i^m,t^{\mathrm{cre}},t^{\mathrm{inv}};r\rangle \wedge i^m\in I^m \wedge r\in R\wedge t^{\mathrm{cre}}\in T\wedge t^{\mathrm{inv}}\in T\}$.

5　面向开发过程的产品状态追踪

5.1　基于时间截面的产品状态追踪

产品结构模型在开发过程中的动态适应性主要体现为多研发主体、多任务分布以及多地域协作开发体系下产品状态的动态组织、动态追踪能力.开发过程中不同状态的产品结构是互相关联、互相约束的,反映了产品结构在开发过程中的时域继承性、延续性,因此产品状态与其时域行为及特征密切相关[3],如图 4 所示.

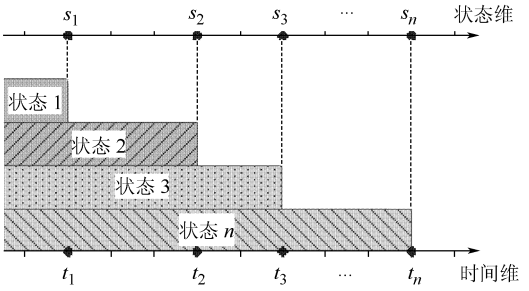

图 4　产品状态的时域行为特征

图 4 可以看出,产品结构沿状态维的延伸与开发任务沿时间维的推进具有映射关系,产品对象内容随开发任务的时域推进不断积累,在状态维则表现为产品状态的动态变迁.产品对象的时域扩展定义(定义 7)包含了产品状态信息(t^{cre},t^{inv}),并通过这种方式记录产品对象的时域行为.实际工程应用中,对产品对象的时域状态标签作如下约定处理:

1) 创建产品对象(属性对象、结构对象)时,生效标签 t^{cre} 设置为当前时间 t_{current}(即时生效),失效标签 t^{inv} 设置为 $+\infty$ 代表(永远不失效);

2) 修改节点对象的特征信息时,先创建该节点的属性对象副本以及该副本和原父节点及所有当前下级节点间的装配关系(结构对象),按照 1)进行状态标签的置位,在此副本上进行特征信息的修改;然后设置原节点对象的失效标签 t^{inv} 为当前时间 t_{current}(当前状态下失效);

3) 删除节点对象时,设置该节点相应结构对象的失效标签 t^{inv} 为当前时间 t_{current},以表明该节点对象当前状态下失效(逻辑删除);

4) 复用一个节点到另一个节点上时,创建被复用节点对应属性对象与目标节点(属性对象)间的装配关系,并按照 1)进行状态标签置位.

可以看出,开发过程中产品结构存在唯一的物理视图,生效状态标签 t^{cre} 和失效状态标签 t^{inv} 在该视图上映射为产品结构随开发进程推进的两条时域线索,产品结构沿着这两条线索不断延拓,反映了产品结构的时域延续特征.产品结构的不同状态视图与时间维上时刻具有映射关系(图 4),可以把时间维上与产品状态对应的时刻 t_s 延伸成一个与时间轴正交的时间截面,进而通过时

间截面实现产品结构状态视图的映射及追踪.

图5可见,时间截面 t_s 分别在生效时间和失效时间线索上形成产品对象的有效区间($-\infty$, t_s]和失效区间[t_s, $+\infty$),采用算法 $t^{cre}\in(-\infty, t_s]\wedge t^{inv}\in(t_s, +\infty)$可以进行产品对象状态视图的映射及关联(实际应用中,t_s 取某一产品状态对应开发阶段的结束时间,如果是当前状态,t_s 则可取当前时间 $t_{current}$).

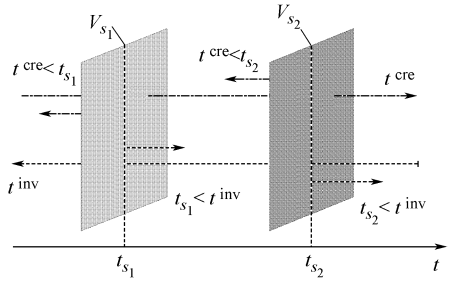

图5 基于时间截面的产品状态追溯

由于产品结构的状态视图(图5中的 V_{s_1}, V_{s_1})来源于唯一的物理视图,物理视图上的产品结构数据通过时间状态标签反映产品结构的时域延续性,所以可以通过产品结构物理视图上的时域特征来管理产品结构不同状态的关联及映射,进而实现产品状态的动态追溯.

5.2 开发过程产品状态追踪实例

此处以一个具体产品结构(局部)在开发过程中的状态变迁实例来说明基于时间截面的产品状态追溯原理(为论述方便,不影响理解的前提下,该例中所有时间参量均取4位整数).

图6所示实例,产品结构从 C(初样)状态变迁到 Z(正样)状态,产品结构作了如下工程调整或修改(V.2→V.4):①节点 N_6(应该为〈N_3, N_6,2〉,为了便于表达和理解,不引起歧义的前提下,简化为 N_6)复用到节点 N_2;②节点 N_6 从原有父节点 N_3 删除;③修改节点 N_3 的特征信息,变为 N'_3.

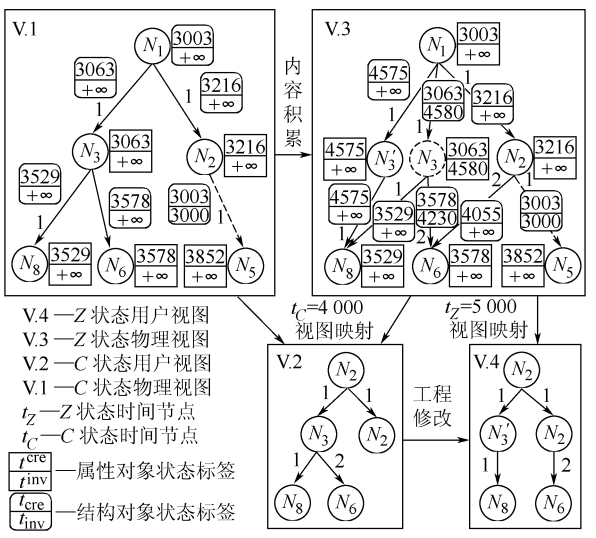

图6 基于时间截面的产品状态追溯实例

上述工程修改映射到物理视图上则表现为产品结构内容的下述积累或延拓(V.1→V.3):①创建结构对象〈N_2, N_6,2〉,设置其生效状态标签 t^{cre} 为当前时间(4 055)、失效状态标签 t^{inv} 为$+\infty$,以表明节点 N_6 即时起复用到节点 N_2;②设置结构对象〈N_3, N_6,2〉的失效状态标签 t^{inv} 为当前时间(4 230),以表明节点 N_6 即时起从节点 N_3 删除;③创建节点 N_3 对应的属性对象副本 N'_3,创建结构对象〈N_1, N'_3,1〉和〈N'_3, N_8,1〉,设置它们的生效状态标签 t^{cre} 为当前时间(4 575),失效状态标签 t 为$+\infty$;设置原节点(属性对象 N_3、结构对象〈N_1, N_3,1〉)的失效状态标签 t^{inv} 为当前时间(4 580);根据实际工程需要修改产品节点 N'_3 的特征信息.

图6中V.3所示物理视图的基础上,可以通过映射算法 $t^{cre}\in(-\infty, t_s]\wedge t^{inv}\in(t_s, +\infty)$进行产品结构不同状态视图的映射,该算法包括两部分:$t^{cre}\in(-\infty, t_s]$是有效性条件,表示所有在时间截面 t_s 之前创建的对象在该状态是有效的,否则无效;$t^{inv}\in(t_s, +\infty)$是失效性条件,表示所有在时间截面之后才逻辑删除(失效)的对象在该状态仍是有效的,否则无效.该例中,C 状态时间截面 $t_C=4\,000$,有效性条件 $t^{cre}\in(-\infty, 4\,000]$表明在 $t_C=4\,000$ 之后创建的对象在 C 状态是无效的,上述工程修改中所有新创建对象(复用节点$\overline{〈N_3, N_6, 2〉}$、修改节点$\overline{〈N_1, N'_3, 1〉}$)均满足条件,故在 C 状态视图上不可见(图6中V.2);失效性条件 $t^{inv}\in(4\,000, +\infty)$表明在 $t_C=4\,000$ 之后才逻辑删除的对象在该状态仍有效,上述工程修改中逻辑删除的对象(节点$\overline{〈N_1, N_3, 1〉}$、$\overline{〈N_3, N_6, 1〉}$)均满足条件,故在 C 状态视图仍然可见(图6中V.2).Z 状态时间截面 $t_Z=5\,000$,有效性条件 $t^{cre}\in(-\infty, 5\,000]$表明 $t_Z=5\,000$ 之前创建的对象在 Z 状态是有效的,上述工程修改中所有新创建对象(复用节点$\overline{〈N_3, N_6, 2〉}$、修改节点$\overline{〈N_1, N'_3, 1〉}$)均满足条件,故在 Z 状态视图可见(图6中V.4);无效性条件 $t^{inv}\in(5\,000, +\infty)$表明在 $t_Z=5\,000$ 之前逻辑删除的对象在该状态是无效的,上述工程修改中逻辑删除的对象(节点$\overline{〈N_1, N_3, 1〉}$和$\overline{〈N_3, N_6, 1〉}$)均满足条件,故在 Z 状态视图不可见(图6中V.4).

可以看出,通过对产品结构定义的时域扩展

(定义 7),可以把针对产品结构的工程修改转化为其物理视图上内容的不断延拓、积累,从物理层面上反映产品结构的时域延续性.产品结构的物理视图在开发过程中是唯一的,不同状态的产品结构从唯一的物理视图上映射而来,保证了开发过程中不同状态产品结构数据的一致性、准确性.可以通过产品结构物理视图的时域延续性来管理产品结构状态之间的关联及约束,进而实现开发过程中产品结构状态的动态追溯.

6　结 束 语

产品数据是连接协同开发过程所有业务活动和动态过程的基础,作为产品数据的核心,合理而高效的产品结构模型是面向开发过程产品数据管理应用的关键.本文提出的产品结构形式化模型及其面向开发过程工程应用的状态追踪方案,在模型层上保证了开发过程产品结构数据不同状态的时域延续性、继承性和一致性,为产品结构模型在实际开发过程中的状态追踪、历史回溯提供了切实可行的实现方法,对于产品结构模型在动态开发环境下的应用具有一定的借鉴意义.

参考文献(References)

[1] Sudarsan R, Fenves S J, Sriram R D, et al. A product information modeling framework for product lifecycle management [J]. Computer-Aided Design, 2005, 37(13): 1399－1411

[2] 孙国梓,郁鼎文,吴志军.面向企业的全局产品结构模型[J].清华大学学报(自然科学版),2004, 44(5): 633－636
Sun Guozi, Yu Dingwen, Wu Zhijun. Enterprise-oriented global product structure mode[J]. Journal of Tsinghua University (Sci & Tech), 2004, 44(5): 633－636 (in Chinese)

[3] Larsen M H, Sϕrensen C, Langer G. Development of a product meta product state model[J]. Computers in Industry,2001, 46:275－287

[4] 吴先超,吕晓枫,孙吉贵.基于广义产品结构的配置和集成研究[J].计算机集成制造系统-CIMS,2006, 12(6): 869－892
Wu Xianchao, Lü Xiaofeng, Sun Jigui. Research on configuration & integration based on generic product structure[J]. Computer Integrated Manufacturing Systems-CIMS, 2006, 12(6): 869－892 (in Chinese)

[5] 黄学文,范玉顺. BOM 多视图和视图之间映射模型的研究[J]. 机械工程学报,2005, 41(4): 97－102
Huang Xuewen, Fan Yushun. Research on bom views and bom view mapping model[J]. Chinese Journal of Mechanical Engineering, 2005, 41(4): 97－102 (in Chinese)

[6] Tang Xiaoqing, Hu Yun. Data model for quality in product lifecycle[J]. Computers in Industry, 2008, 59(2/3):167－179

[7] Lee G H, Sacks R, Eastman C. Product data modeling using GTPPM-A case study [J]. Automation in Construction, 2007, 16:392－407

[8] 范玉青.现代飞机制造技术[M].北京:北京航空航天大学出版社, 2001
Fan Yuqing. Modern aircraft manufacturing technology [M]. Beijing: Beijing University of Aeronautics and Astronautics Press, 2001 (in Chinese)

[9] 段桂江,孙飞.基于功能网链模型的企业应用系统融合分析方法[J].计算机集成制造系统-CIMS,2006,12(1):65－70
Duan Guijiang, Sun Fei. Function-network-based method analyzing integration of enterprise application systems[J]. Computer Integrated Manufacturing Systems-CIMS, 2006, 12(1):65－70 (in Chinese)

[10] 樊蓓蓓,祁国宁.基于复杂网络的产品族结构建模及模块分析方法[J].机械工程学报,2007,43(3): 187－198
Fan Beibei, Qi Guoning. Modeling of product family structure and module analysis method based on complex network [J]. Chinese Journal of Mechanical Engineering, 2007, 43(3): 187－198(in Chinese)

(责任编辑:文丽芳)