

语义特征造型的拓扑元素编码机制*

高雪瑶, 孙立镭

(哈尔滨理工大学 计算机科学与技术学院, 哈尔滨 150080)

摘要: 针对现有拓扑元素标志与编码方法都是面向基于历程的造型系统, 无法完全满足与历程无关语义特征造型的需要这一问题, 提出一种以特征为基础的拓扑元素标志方法, 并归纳出了一种统一格式的拓扑元素编码方法。当拓扑元素发生分裂时, 给出了区分分裂面和分裂边的方法及编码格式; 根据模型修改前后拓扑元素的不同变化情况, 提出虚拓扑元素和子边的概念及编码格式, 通过保留模型中拓扑元素间的拓扑关系, 实现与历程无关的模型修改操作, 并将该方法在自主开发的语义特征造型系统(HUST-CAID)中加以实现。

关键词: 与历程无关; 拓扑元素; 编码; 虚拓扑元素; 子边

中图分类号: TP391 **文献标志码:** A **文章编号:** 1001-3695(2010)03-0938-03

doi:10.3969/j.issn.1001-3695.2010.03.035

Coding mechanism for topological entities in semantic feature modeling

GAO Xue-yao, SUN Li-juan

(School of Computer Science & Technology, Harbin University of Science & Technology, Harbin 150080, China)

Abstract: The naming and coding for topological entities were all for history-based modeling systems, which could not entirely support history-independent semantic feature modeling. This paper proposed a feature-based method for naming topological entities, and proposed a uniform coding method of topological entities. Gave differentiate methods and coding format for the split faces and edges. And proposed virtual topological entity and sub-edge according to topological entities variation after model modifying. Remained relations between topological entities to realize history-independent model modification. The proposed method was realized in HUST-CAID.

Key words: history-independent; topological entity; coding; virtual topological entity; sub-edge

拓扑元素的命名与编码是参数化造型系统的关键技术^[1], Bronsvoort 等人^[2]把它归结为目前参数化特征造型系统需要解决的六大问题之一。Kripac^[3]设计了一个拓扑标志系统, 把被引用的拓扑元素 ID 从拓扑标志系统映射到新模型中的相应元素。苏晓锋等人^[4]提出一种面和边的编码方法, 并按照子面参数域在父面参数域上的顺序对子面排序以区分分裂面。陈正鸣等人^[5]利用面的方向性定义了三类面面进出关系, 并以此表示拓扑元素的局部拓扑信息。Zheng 等人^[6]用步骤 ID 和内部 ID 命名面, 用子面的邻接面区分分子面, 并用邻接面命名边和顶点。Yan 等人^[7]提出一种基于几何连续性的参数族确定方法, 并以此建立几何和拓扑元素在新旧模型中的映射方法。Wang 等人^[8]定义了名字传递图(NPG)来实现拓扑实体名的一致性传递, 给出了辨识初始拓扑实体的规则与算法。Wu 等人^[9]提出一种基于图的拓扑元素编码与解码方法, 并结合参数化信息实现新旧模型的相似性匹配。

以上方法都是针对基于历程的造型系统, 而语义特征造型不基于设计历程, 是造型系统一个新的发展方向, 上述方法无法完全满足语义特征造型的需要。在与历程无关的语义特征造型系统中修改模型, 只需在现有模型上进行操作, 无须根据设计历程重构整个模型。本文提出一种基于特征的拓扑元素标志与编码方法, 并根据与历程无关模型修改的动态、可逆的

特点, 提出子边和虚拓扑元素的概念及编码方法, 以实现与历程无关的模型修改操作。

1 拓扑元素的标志

三维空间中的直线可用两个相交的面来表示, 点可用三个相交于一点的面来表示。将这一思想应用到拓扑实体的标志与编码上, 拓扑边通过两个邻接面的组合来表示, 拓扑点通过三个邻接面的组合来表示。具体方法为: 以特征为基础, 首先对特征实体的面进行标志与编码, 然后利用面的拓扑名对边和点进行标志与编码。

1.1 面标志

基本形状特征主要由扫略操作和旋转操作生成, 具有可明确定义的面。本文规定其各个面的标志规则为: 以特征名和面在特征内部的名字共同确定。当特征依附于某一个基准面(依附面)时, 规定与该基准面相接的面为特征的底面, 与之相对的面是顶面。描述规则为(feature_instance_name, feature_element_ID)。

由扫略和旋转操作生成基本形状特征时, 可能生成多个侧面, 需要对这些侧面进行区分。扫略操作分为平扫(拉伸)和广义扫。平扫生成特征侧面的数量与草图轮廓边的数量相等, 可将各个侧面的编号直接对应于草图边的编号。广义扫的路

收稿日期: 2009-08-14; 修回日期: 2009-09-23 基金项目: 国家自然科学基金资助项目(60173055)

作者简介: 高雪瑶(1979-), 女, 黑龙江哈尔滨人, 博士, 主要研究方向为语义特征造型、约束求解(gaoxueyao@hotmail.com); 孙立镭(1944-), 男, 黑龙江哈尔滨人, 教授, 博导, 主要研究方向为计算机图形学与 CAD。

径是二维或三维连续线段,每一条草图边对应的侧面数可以是多个。因此先给草图轮廓边分配 ID 编号,然后对扫描路径 P 进行分段,最后依据路径段号 P 和轮廓 ID 号对各个侧面进行标志。描述规则为 (sweep. side_P_ID)。对于旋转操作生成的特征,首先给各轮廓边编号,判断各轮廓边的起点到旋转轴的距离,将具有最小值的边记为基准边,将下一条边的 ID 记为 1,其他后续各边 ID 依次加 1,然后根据编号对各旋转侧面进行标志。描述规则为 (rotate. side_ID)。

当特征面分裂为若干个子面时,需要对这些子面加以区分。本文借鉴文献[4]的方法并加以改进:按照子面在父面参数域上的顺序对子面进行排序,并保留父面 ID,用于说明拓扑元素的变化情况。假设特征所在局部坐标系为 $u-v-w$ 坐标系,首先对父面进行参数化,然后求出各个子面在父面上的参数域,按照先 u 后 v 最后 w 方向的顺序对各个子面排序。描述规则为 (feature_instance_name. feature_element_ID, f_ID)。在图 1 中,block 特征的顶面被两个 slot 特征分裂为四个子面,它们的父面 ID 即 feature_instance_name. feature_element_ID 都是 block. top,子面 ID 即 f_ID,分别为 f_1 f_2 f_3 f_4 。父面 block. top 为平面,因此忽略子面在 w 方向上的排序。

1.2 边标志

特征相交、修改等操作都会引起边的变化。对边的频繁引用要求正确区分各条边。拓扑边可由两个邻接面惟一确定,边的端点由另一个面与这两个相邻面相交形成,因此可以通过由特征定义的拓扑面来标志拓扑边。描述规则为 (adjacent_feature_faces), (end_faces)。其中,adjacent_feature_faces 为边的两个邻接面,end_faces 为边的两个端点所在的起始面和终止面。例如图 2 中的 e_1 边表示为 (block. top, block. side₁), (block. side₄, blindslot. side₁)。

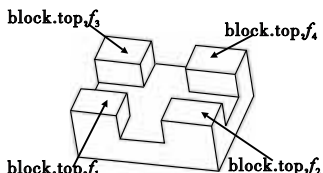


图1 分裂子面的标志

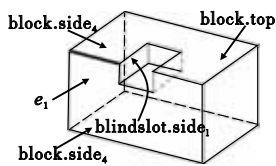


图2 分裂边的标志

当模型中存在曲面时,两个面相交产生的边可能多于一条。图 3 中圆柱体的顶面与圆槽的侧面相交生成两条边,这两条边的 adjacent_feature_faces 都是 cylinder. top 和 roundslot. side,end_faces 都是 cylinder. side,无法正确区分。这种情况可以通过参考线 reference 来解决:定义一条有方向的参考线,规定沿参考线的方向前进,行进方向的左侧区域为正,以此将需要区分的拓扑元素划分在参考线的两侧,以进行识别。描述规则为 (adjacent_feature_faces), (end_faces, reference. orientation)。其中,reference. orientation 用来说明要区分的拓扑元素位于参考线的哪一侧。对于图 3 中的两条边,通过定义一条位于 cylinder 的顶面且与两条相交边平行的参考线 reference₁ 来区分:

$$e_1 : (\text{cylinder. top, roundslot. side}), (\text{cylinder. side, roundslot. side}, \text{reference}_1. \text{positive})$$

$$e_2 : (\text{cylinder. top, roundslot. side}), (\text{cylinder. side, roundslot. side}, \text{reference}_1. \text{negative})$$

以上例子中,定义的参考线是直线。图 4 中,在长方体的

顶面开一个环形槽,形成内外两条封闭环形边,要区分这两条边需要定义一条封闭的曲线参考线:令参考线通过环形槽的中心线。因此,两条环形边可以分别表示为

$$e_1 : (\text{block. top, roundslot. side}), (\text{reference. positive})$$

$$e_2 : (\text{block. top, roundslot. side}), (\text{reference. negative})$$

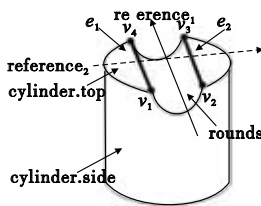


图3 利用参考线标志边

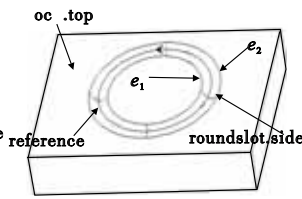


图4 用封闭参考线区分边

由于这两条边为封闭边,没有起点与终点,end_faces 忽略不计。

这里给出定义参考线的原则:参考线(包括直线和曲线)必须与特征的某些基准面、基准轴线或对称面相关联,如通过或垂直于轴线、与基准面重合或垂直等;参考线与特征的某条基准线确定一个平面,使要区分的两条边在该平面上的投影落在参考线的两侧;在该平面上,沿参考线的方向前进,行进方向的左侧区域为正;当特征的位置发生变化时,参考线的位置也相应地发生变化。

1.3 点标志

在实体的拓扑关系中,顶点可看成面的交点,它的邻接面集刻画了其拓扑局部性质。因此顶点可以由其邻接面集来表示,如图 5 中的点 v_1 为 (block. top, slot. side₂, block. side₁)。

在有些情况下,仅由邻接面集无法惟一确定顶点,如图 3 中的边 e_1 、 e_2 的四个顶点的邻接面都为 cylinder. top, roundslot. side 和 cylinder. side。这种情况可借助参考线来区分,描述规则为 (adjacent_feature_faces), (reference. orientation)。在图 3 中定义一条与 reference₁ 垂直的参考线 reference₂ (图 3 中虚线所示),因此四个顶点分别表示为

$$v_1 : (\text{cylinder. top, roundslot. side, cylinder. side}), (\text{reference}_1. \text{positive}, \text{reference}_2. \text{negative})$$

$$v_2 : (\text{cylinder. top, roundslot. side, cylinder. side}), (\text{reference}_1. \text{negative}, \text{reference}_2. \text{negative})$$

$$v_3 : (\text{cylinder. top, roundslot. side, cylinder. side}), (\text{reference}_1. \text{negative}, \text{reference}_2. \text{positive})$$

$$v_4 : (\text{cylinder. top, roundslot. side, cylinder. side}), (\text{reference}_1. \text{positive}, \text{reference}_2. \text{positive})$$

2 拓扑元素的编码

在与历程无关的造型系统中,设计人员对模型进行修改,只需在现有模型上进行操作和编辑,而无须根据设计历程重构整个模型。例如将图 6 中的圆槽向右拖动,使 e_1 与 e_2 边重合。该操作使长方体顶面分裂而成的子面 f_2 消失。若再次修改模型,将圆槽移回原处,则 f_2 应被正确恢复。

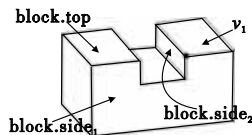


图5 点的命名

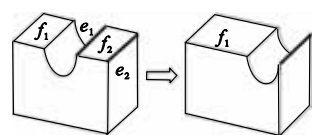


图6 特征修改:向右移动圆槽

通过上述例子可以看出,与历程无关的模型操作是动态

可逆的。在模型修改过程中,会出现某些拓扑元素互相重合或消失的情况,随着进一步的操作,这些元素可能再次出现。因此,与传统的基于历程的造型方法不同,对操作过程中消失或重合的拓扑元素不能作简单的删除或合并,而要保留初始模型中特征拓扑元素间的拓扑关系,不改变特征所拥有的属性。为了实现上述操作过程,根据前面分析的拓扑元素标志方法,本文归纳出一种统一格式的拓扑元素编码方法,并根据模型修改前后拓扑元素的不同变化情况,提出虚拓扑元素和子边的概念,以实现与历程无关的模型操作过程。

2.1 面与虚面的编码

根据拓扑面的标志方法,将面的编码格式统一表示为

$$\text{face_ID} = (\text{featureID}, \text{type}, \text{index}, \text{PID}, \text{split}, \text{addi})$$

FeatureID 表示该面所属的特征的标号。

Type 表示面的类型,其值为 -1 表示底面,0 表示顶面,1 表示由扫略操作生成的侧面,2 表示旋转操作生成的侧面。

当有多个连续侧面时,用 index 记录各个侧面的编号。对于底面和顶面,index 的值为 0。

当 type 为 1 时,用 PID 表示扫略特征的路径编号。当 PID 的值为 1 时,表示拉伸或广义扫的第一段路径;当大于等于 2 时,表示广义扫后续各段路径的编号。对于底面和顶面,PID 为 0。当 type 为 2 时,该面为旋转侧面,此时 PID 用来记录旋转角度。

Split 表示面是否发生了分裂,其值为 0 表示没有发生分裂,为 1 表示发生分裂。当面发生分裂时,置其 split = 1,并按照前述方法对分裂子面进行排序。为了避免出现编码位空闲而导致过多空间浪费的情况,本文为分裂的子面重新分配空间并进行编码,形成面分裂表 faceSplitTable。子面编码格式为 subface_ID = (featureID, type, index, PID, split, fID)。其中, fID 表示子面的编号,其余各项的含义与前面所述相同,用于惟一表示子面的父面。

Addi 为附加位,用于处理在模型修改过程中消失或重合的拓扑元素。对于这类拓扑元素,本文不将其删除或合并,而是将其转换为虚拓扑元素,保留其引用和拓扑关系,不显示在模型的几何边界上。用 addi 位处理这种情况,addi 的值为 1 表示该面构成模型的几何边界,为 0 表示消失或与其他面重合,即该面为虚面。虚面表示为 vFace_ID = (featureID, type, index, PID, split, addi)。此时 addi 的值为 0。

当拓扑面分裂为若干子面时,将该面的标志号转换为虚标号,并生成子面编码。若子面在操作过程中消失,则将该子面编码转换为虚标号,表示为虚子面: vsubface_ID = (featureID, type, index, PID, fID, addi)。此时 addi 的值也为 0。

2.2 边与虚边的编码

根据拓扑边的标志方法,将边的编码格式统一表示为

$$\text{edge_ID} = (\text{adjFaceIDs}, \text{endFaceIDs}, \text{RefOriCode}, \text{addi})$$

AdjFaceIDs 表示生成此边的两个邻接面。

EndFaceIDs 表示边的两个端点所在的起始面和终止面。当所区分的边为环形封闭边时, endFaceIDs 的值为 0。

RefOriCode 用于区分该边位于参考线的哪一侧,当它的值为 -1 时,表示沿参考线方向行进时,该边在参考线的右侧,为 1 表示在左侧。无须用参考线区分时, refOriCode 的缺省值为 0。

Addi 的含义与面编码中的含义相同。

当被引用的边消失或不构成模型的几何边界时,将该边表示为虚边: vEdge_ID = (vadjFaceIDs, endFaceIDs, refOriCode, addi)。其中, vadjFaceIDs 表示该虚边的两个邻接面,且至少有一个邻接面是虚面; addi 的值此时置 0。

2.3 点与虚点的编码

根据拓扑点的标志方法,将点的编码格式统一表示为 vertexID = (adjFaceIDs, refOriCode)。其中, adjFaceIDs 表示点的三个邻接面; RefOriCode 的含义与边编码中的含义相同,缺省值为 0。

虚拓扑点表示为 vertexID = (vadjFaceIDs, refOriCode, addi)。其中, vadjFaceIDs 表示该虚点的三个邻接面,且其中至少有一个邻接面是虚面;此时 addi 的值为 0。

2.4 子边及其编码

模型修改过程中,若有拓扑元素发生重合,可能出现将一条边划分为若干段,而其中几段不构成模型几何边界的情况。对此,本文提出子边的概念:属性相同的拓扑元素(主要是面和边)重合,使拓扑边上生成新拓扑点,这些拓扑点将拓扑边划分为几段,每一段都是该拓扑边的一段子边。

当模型中存在重合面或重合边时,子边的生成方法如下:

a) 求重合拓扑元素(面或边)在几何模型上的交集,记录构成该交集的拓扑边及边的端点。

b) 判断哪些端点分别位于原拓扑元素的互相相交或重合的边上。

c) 用这些端点对所在拓扑边进行分段,每一段都是原拓扑边的一段子边。位于交集内部的子边标号为虚标号。

子边的编码格式为 subEdge_ID = (edgeID, adjFaceIDs, vendFaceIDs, addi)。其中, edgeID 为子边所属拓扑边的标号; adjFaceIDs 表示该子边的两个邻接面; vendFaceIDs 为子边的起止面,且其中至少有一个来自于与该边重合的拓扑边的起止面; addi 的含义与面、边编码中的含义相同。

位于交集内部的子边标号为虚标号: vsubEdge_ID = (edgeID, vadjFaceIDs, vendFaceIDs, addi)。其中, vadjFaceIDs 为子边的两个邻接面,且至少有一个邻接面位于交集的内部。该邻接面的拓扑边界为生成子边时所求的围成该交集的拓扑边,此邻接面和它的拓扑边均不构成模型的几何边界,标号记为虚标号,且 addi 的值为 0。

3 方法实现与实例

将本文提出的拓扑元素编码机制应用到自主开发的语义特征造型系统(HUST-CAID)中,系统采用 VC++ 6.0 加以实现,将编码作为属性与拓扑元素联系起来。在特征建模及修改过程中,当拓扑元素发生分裂或重合时,对其进行重新编码,并用面分裂表 faceSplitTable 和边分裂表 edgeSplitTable 记录发生分裂的拓扑元素。图 7 是初始模型,在基体 block 上开两条通槽 slot₁ 和 slot₂。其中 slot₁ 是圆槽。在 slot₂ 上是一个圆柱 cylinder,且不与 slot₁ 相交。对模型进行修改,减小 slot₂ 的深度,令它与圆柱相交,产生两条平行交线,并对其中一条进行倒圆角操作。利用本文的拓扑元素标志与编码机制可正确区分这两条边,模型修改结果如图 8 所示。

性而获得更多的构件信息。

b)支持不同构件生产者的个性化描述。构件生产者可以自主、灵活地管理属于自己的特殊属性描述,并对自己的构件进行个性化描述,使其发布的构件更受使用者的欢迎。

c)较好地支持构件的各种查询方式并扩展了查询范围。该方法可以支持一般用户的简单查询和特殊用户的特殊属性查询。用户在查询时,可以以通用属性的剖面作为查询条件,也可以将特殊属性的剖面作为查询条件进行更深层次的查询。在查询特殊属性时,用户可以通过该生产者的特殊属性的定义和其所有可见构件来学习该生产者生产的构件特性以及其生产经验。企业在复用构件时,构件查询的范围可扩大到整个分布式构件库。

d)易于实现构件库的维护和管理。构件通用属性和特殊属性分开后,管理员只需维护构件的通用属性,对特殊属性只需作一些必要的限定即可。特殊属性将由生产者自行维护和管理,从而在一定程度上减轻了管理员的工作。

4 结束语

CGADM 通过对现存构件描述方法提出的剖面分类分析,得到一种比较完善、通用、高层次的构件通用属性的描述方法。它的主要思想就是将构件的通用属性和特殊属性分开描述和管理,一方面能够较好地提高构件描述的准确性和规范性,另一方面也可以满足不同构件生产者对构件个性化的描述需求。该方法同智能主体和术语扩展的思想相结合,将能够更好地优化和完善构件库的构件分类和管理,同时也适用于分布式构件库系统。

(上接第 940 页)然后拖动圆柱修改其位置,则原来被倒角的边随着圆柱位置的改变而消失,倒角也随之消失。操作结果如图 9 所示。

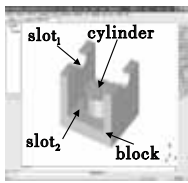


图7 初始模型

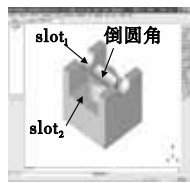


图8 减少slot2的深度并倒圆角



图9 修改cylinder的位置

4 结束语

语义特征造型是与历程无关的造型方法,其建模与修改过程不依赖于造型历程,具有操作灵活方便的特点,是造型系统的一个新的发展方向,具有良好的发展与应用前景。拓扑元素的命名和编码是系统实现的一个关键问题,本文分析了现有拓扑元素命名方法存在的主要问题,提出了一种支持与历程无关特征造型的拓扑元素标志与编码方法。通过保留模型中拓扑元素间的拓扑与引用关系,避免了在基于历程造型中存在的拓扑元素匹配问题,实现了在模型修改过程中对拓扑元素的唯一标志与编码。根据特征模型修改前后拓扑元素的变化情况,提出虚拓扑元素和子边的概念,以实现与历程无关的模型修改操作,并将该方法在自主开发的语义特征造型系统中加以实现。

参考文献:

[1] 荆树旭,何发智,刘华俊. 拓扑元素永久命名综述[J]. 计算机

参考文献:

[1] 汤敏. 基于构件通用属性描述语言的构件注册与检索[D]. 武汉: 武汉理工大学, 2008: 10-12.

[2] 袁兆山, 王淑红. 可复用构件表示模型研究[J]. 计算机应用, 2001, 21(1): 1-3.

[3] WU Qiong, CHANG Ji-chun, MEI Hong, et al. JBCDL: an object-oriented component description language [C]//Proc of Technology of Object-Oriented Languages and Tools. 1997: 198-205.

[4] SIDNEY C B. Software reuse as ontology negotiation [C]//Proc of the 8th International Conference on Software Reuse. 2004: 242-253.

[5] 马亮, 孙艳春, 孙家. 一个可扩展的 OO 构件模型的设计[J]. 小型微型计算机系统, 2001, 22(10): 1202-1205.

[6] 钟林辉, 谢冰, 邵维忠. 扩充 CDL 支持基于构件的系统组装与演化[J]. 计算机研究与发展, 2002, 39(10): 1361-1365.

[7] TENG Qi-ming, CHEN Xiang-qun. XCODE: an extensible component description language for system software [J]. Acta Scientiarum Naturalium Universitatis Pekinesis, 2004, 40(3): 388-396.

[8] 谭建, 王平涛, 许满武. WACDL: 基于 XML 的 Web 应用构件描述定义语言[J]. 计算机应用研究, 2004, 21(9): 98-101.

[9] 马锐. 基于剖面分类模式的构件检索技术研究[D]. 大连: 大连海事大学, 2008: 18-19.

[10] Reuse Library Interoperability Group. RIG uniform data method for reuse libraries(UDM), RPS-0002[R]. 1994.

[11] 李晓丽, 刘超, 金茂忠, 等. 《软件构件产品质量》标准介绍[J]. 信息技术与标准化, 2006(6): 19-24.

[12] 赵俊峰. 软件构件标准概述[J]. 信息技术与标准化, 2006(6): 10-13.

[13] 常继传, 李克勤, 郭立峰, 等. 青鸟系统中可复用软件构件的表示与检索[J]. 电子学报, 2000, 28(8): 20-23.

[14] 薛云皎. 基于智能主体的分布式构件库相关技术研究[D]. 上海: 复旦大学, 2006: 16-25.

辅助设计与图形学学报, 2007, 19(5): 545-552.

[2] BRONSVOORT W F, BIDARRA R, NYIRENDA P J. Developments in feature modelling[J]. Computer-Aided Design & Applications, 2006, 5(3): 655-664.

[3] KRIPAC J. A mechanism for persistently naming topological entities in history-based parametric solid models [J]. Computer-Aided Design, 1997, 29(2): 113-122.

[4] 苏晓锋, 黄正东, 朱林, 等. 形状特征中的拓扑元素编码体系[J]. 计算机辅助设计与图形学学报, 2000, 12(2): 137-141.

[5] 陈正鸣, 高曙光, 张凤军, 等. 一种拓扑元素的命名和辨识方法[J]. 计算机学报, 2001, 24(11): 1270-1277.

[6] ZHENG Jia-jun, FAN Ming, TONG Ruo-feng, et al. A mechanism for persistently naming topological entities [C]//Proc of the 8th International Conference on Computer Supported Cooperative Work in Design Proceedings. New York: IEEE, 2004: 84-89.

[7] WANG Yan, NNAJI B O. Geometry-based semantic ID for persistent and interoperable reference in feature-based parametric modeling[J]. Computer-Aided Design, 2005, 37(10): 1081-1093.

[8] WANG Yan-wei, WU Jun-jun, CHEN Li-ping, et al. Identity propagation method for tracing alterations of a topological entity in a history-based solid modeling system [J]. Journal of Advanced Manufacturing Technology, 2005, 27(3-4): 305-312.

[9] WU Tao, XI Yi-shan, LI Zhang, et al. Coding and decoding of topological entities in constraint-based variational design [C]//Proc of International Conference on Advances in Product Development and Reliability. 2008: 201-208.