

面向特征的产品数据库中特征有效性的维护

唐少辉¹ Ma, Y. S² Chen, G.³

(¹ 广东工业大学 机电工程学院, 广州 510090)

(² 加拿大阿尔伯塔大学 机械系, 爱德蒙顿 AB T6G 2G8)

(³ 新加坡南洋理工大学 机械与宇航工程学院, 新加坡 639798)

Maintaining feature validity in feature-oriented product database

TANG Shao-hui¹, Ma, Y. S², Chen, G.³

(¹Guangdong University of Technology, School of Mechanical and Production Engineering, Guangzhou 510090, China)

(²University of Alberta, Department of Mechanical Engineering, Edmonton AB T6G 2G8, Canada)

(³Nanyang University of Technology School of Mechanical and Aerospace Engineering, Singapore 639798)

【摘要】并行和协同产品设计已经成为当前的流行设计模式。具有粒度的产品数据库是理想的支持基于网络的协同设计数据源。重点讨论如何维护面向特征的数据库中产品特征模型的有效性。应用所提出的约束解决方法,能保证产品数据库中特征模型的有效性以支持多应用产品协同设计。

关键词: 面向特征; 数据库; 特征有效性; 约束条件; 协同工程

【Abstract】 Concurrent and collaborative engineering (CCE) has become an industry trend. A fine-grain feature-oriented product database is ideal to support web enabled collaborative engineering services. It describes the maintenance of feature validity in the feature-oriented database on the basis of previous work. Under the proposed constraint solving methods, feature information, which includes feature geometry and constraints, can be well managed and thus collaborative product design among multiple applications can be achieved.

Key words: Feature-oriented; Database; Feature validity; Constraint; Collaborative engineering

中图分类号: TP391.72, TP311 文献标识码: A

随着全球范围内市场竞争的加剧,并行和协同工程已经成为产品设计的流行模式。传统的产品数据库一般只保存产品模型的几何信息,且以文件的形式存储。在研究工作中,具有粒度的面向特征的产品数据库被用来支持多应用协同产品设计。数据库中,产品信息(几何信息和非几何信息)以数据颗粒的形式分布在各个数据表格中。这样的数据结构可以自然、灵活的支持协同设计所需要的多视角。

研究工作表明,具有粒度的面向特征的数据库可以理想的支持基于网络的协同产品设计。目前产品设计基本上都采用基于特征的设计方式,因此在信息共享过程中特征信息的保存非常重要,特征信息的丢失会导致设计的修改无法进行。本文着重研究在具有粒度的面向特征的产品数据库中,如何保证模型特征信息的有效性,以支持多应用产品协同设计。

1 特征几何信息的校验

特征的有效性: 当一个特征满足所有相关的约束条件(本地约束或者跨应用约束),且特征的几何信息有效时,该特征是有效的特征。

特征信息有效性的验证主要包括两方面,即特征几何信息的校验和约束条件的解决。数据库中特征有效性的校验方法,如图 1 所示。

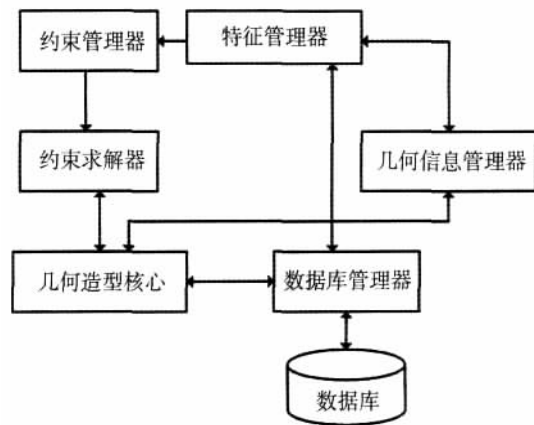


图 1 特征信息的有效性验证

Fig.1 Feature validity maintenance

特征几何信息校验有几何信息管理器来控制。校验过程中,几何信息管理器通过控制一个整合进来的几何造型核心来具体进行验证。几何造型核心(ACIS)在过程中提供底层的几何造型功能。

例如: is_FACE() 可以验证是否是一个有效的面。这里应用几何造型核心可以有效的降低开发工作的难度。

* 来稿日期: 2007-10-30

2 约束条件的解决

所有的约束都由约束管理器以约束图的方式来管理, 解决约束的过程包括两部分, 即本地的约束解决和跨应用约束解决。

2.1 本地约束的解决

在前面的研究工作中^[2], 我们定义了四种约束, 即几何约束, 代数约束, 尺寸约束和语义约束。几何约束定义模型特征元素之间的几何关系, 如图 2 所示。例如共面, 共线, 共点。代数约束用等式来定义特征参数之间的关系, 如孔的直径等于圆柱直径的一半。尺寸约束定义特征元素之间的距离。语义约束定义特征元素的拓补关系。如图 2 所示。例如通孔的语义约束为两个端面不在模型边界上且圆柱面至少部分在模型边界上。

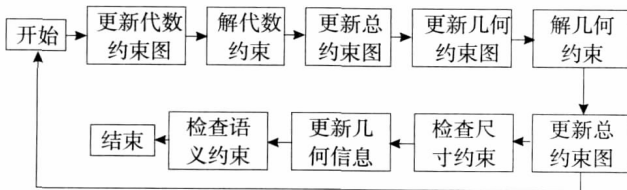


图 2 本地约束解决方案
Fig.2 Methods for local constraint solving

为减少程序开发的难度, 本研究中应用 Dahmen^[3]的方法, 即用不同的约束求解器来解决不同的约束。应用 SkyBlue^[4]来解决代数约束。应用 DOFA 来解决几何约束。本地约束的解决过程。具体如下: (1) 更新代数约束图。代数约束图是总约束图的子图。更新时要加入新的代数约束, 或者把参数传递给修改过的代数约束。(2) 用 SkyBlue 解决代数约束。并用解决好的代数约束子图更新总约束图。(3) 用更新的总约束图得到几何约束子图, 应用 DOFA 方法解决几何约束, 并用新解得的几何约束图来更新总约束图。(4) 根据尺寸约束检查特征参数值, 并根据特征参数更新模型几何信息。(5) 检查特征的语义约束。

由于有些特征参数是隐性参数, 所以上述过程可能有一次重复, 在图 2 中表示为返回起点的箭头。即在第一次解代数约束时, 只有含有完全显性参数的代数约束可解。在几何约束求解结束后, 当所有的隐性参数都成为显性参数时, 原来含有隐性参数的代数约束才可解。这时更新总约束图并返回开始重复一次。

对于语义约束的检查, 需要进行面的边界分析。这里我们应用文献中的边界分析方法。

2.2 跨应用约束的解决

在解决本地约束之后, 需要把改变传递到其它相关的应用中。这种传递可以通过定义跨应用约束来实现。跨应用约束在几何信息的基础上来建立产品模型多应用之间的联系。跨应用约束可以在一般约束模型基础上定义。除了继承一般约束模型的所有属性, 跨应用约束需要定义 linked_views 来指明所连接的不同应用。跨应用约束同样可以有多种, 如共面, 共线和共点等。

在本研究所提出的面向特征的产品数据库中, 所有跨应用约束都通过跨应用约束表来进行管理, 定义 (见表 1)。不同种类的约束的具体内容在该种约束的表中进行操作, 并通过具体约束的 ID constraint_id 来进行识别。

在本地约束的解决过程中, 跨应用约束处于非激活状态。当本地约束求解完成, 且该应用在数据库中更新后, 才会触发数据

库的程序来激活跨应用约束。数据库中的触发器针对行对象进行设计。跨应用约束在数据库中的解决方案如下: (1) 如果没有跨应用约束, 返回; (2) 如果有跨应用约束, 但是当前模型没有修改, 返回; (3) 如果有跨应用约束, 且该约束的唯一参照 ID 没有发生改变, 则向所连接的应用传递改变 (通过 Linked_view_list), 并在该应用中重新求解约束以满足跨应用约束所传递的改变; (4) 如果有跨应用约束, 且该约束的唯一参照 ID 发生改变, 数据库触发程序通知该约束所连接的应用根据数据库中的最新模型来更新该应用中的模型。

表 1 跨应用约束表

Tab.1 Inter-application constraint management

表列	数据形式	说明
Constraint_id	OID	跨应用约束的 ID
Owner_list_id	REF	指向物主的 ID 参照
Linked_view_list	Varchar2	连接的不同应用
Constraint_type	Varchar2	约束的形式

2.3 约束冲突的解决

本研究中通过下述两种方式来处理约束冲突。

2.3.1 约束力- constraint_strength

为处理过约束情况, 在约束中定义了约束力, 即约束的强弱。约束力分为必需, 强, 中和弱等级。约束求解器 SkyBlue 在遇到约束冲突时, 会自动将最弱的约束取消。当有两个等级相同的弱约束时, SkyBlue 会任意取消其中之一。

2.3.2 域力- domain_strength

域力是在发生跨应用约束冲突时, 定义不同应用的强弱 (域力强的应用的约束优先满足)。本文不讨论欠约束情况。

3 实例

实例模型, 如图 3 所示 (初始设计)。约束图实例略。

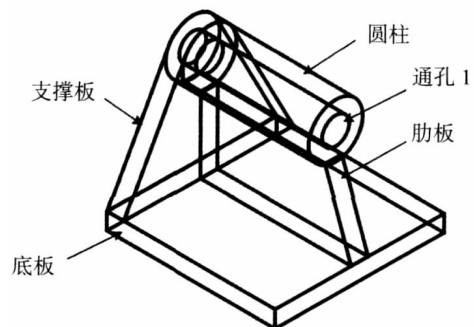


图 3 实例模型
Fig.3 An example part

本文所提出的解决约束方法被应用在一个面向特征的分布式协同设计环境中。数据库服务器 ORACLE9i 在南洋理工大学 CAD/CAM 实验室, 应用服务器和客户端在 ADAM (advanced design and modeling) 实验室。客户端和服务器之间通过传递操作来减少数据传输量。系统说明请参照文献^{[1][2]}。

在以后的设计中, 加入了两个新的特征, 即圆柱突台和通孔。加突台时, 只需解决几何约束。Offset 约束确定了突台上表面的 Z 坐标并确定三个转动自由度, 两个 distance 约束确定了上表面的 X 和 Y 坐标。加通孔时, 代数约束决定了孔径。然后在

基于网络智暴法、并行工程和魔球理论的 汽车开发方法研究

丁志华¹ 代红梅² 黄强¹

(¹九江学院 机械工程学院, 九江 332005)(²九江学院 商学院, 九江 332005)

Research on development method of automobile based on net brainstorm
concurrent engineering and magic ball theory

DING Zhi-hua¹, DAI Hong-mei², HUANG Qiang¹

(¹Mechanical Engineering College, Jiujiang 332005, China)

(²Commercial Science College, Jiujiang University, Jiujiang 332005, China)

【摘要】提出了基于网络智暴法、并行工程和魔球理论的汽车开发方法。在该方法中开发准备、汽车设计、汽车制造和汽车营销四个模块汽车开发为中心构成一个魔球。在各项开发工作中,网络智暴法都是主要的创新方式。

关键词: 网络智暴法; 并行工程; 魔球理论; 汽车开发

【Abstract】 The development method of automobile is put forward which is based on Net Brainstorm, Concurrent Engineering and Magic Ball theory. There are four modules such as development preparation, automobile design, automobile manufacturing and automobile sell which make a magic ball. Net Brainstorm is main innovation mode in every job of automobile development.

Key words: Net brainstorm; Concurrent engineering; Magic ball theory; Automobile development

中图分类号: TH12,TP319 文献标识码: A

汽车产品由一万多个零部件组成,其产品开发是一项复杂的系统工程。受观念影响和设计手段的限制,传统的汽车产品开发过程采用串行方式进行。由于其在设计阶段不能有效地反馈

汽车产品的可制造性、可装配性、可维护性以及产品质量等诸多信息,造成产品在制造阶段或安装阶段才有可能发现问题,从而使汽车产品开发周期长、成本较高、质量难以得到保证。为适应

* 来稿日期: 2007-10-15

更新总约束图后,通过解几何约束,即图 4 中共面(coplanar)和同心(concentric)约束,可以得到孔的定位。此时,再沿图 3 中的返回箭头重新解代数约束(algebraic constraint)来确定孔的高度。最后检查其它的约束并更新模型。更新的模型,如图 4 所示。注意由于研究中采用单元模型(详情请参考文献^[9])作为几何模型,故图 3 和图 4 所显示的是零件的单元模型,而非边界模型。

4 结论

本文在以前研究工作的基础上,着重分析了具有粒度的面向特征的数据库中特征模型有效性的验证方法。研究中采用不同的约束求解器来解决不同的约束。通过定义和解决跨应用约束建立不同应用之间的联系,从而为多应用产品协同设计实现奠定基础。

参考文献

- 1 Tang S H, Ma Y S and Chen G, A feature-oriented database framework for web-based CAx applications [J], Computer-Aided Design & Applications, 2004(1): 117-125
- 2 Ma Y S, Tang S H and Chen G, A Fine-grain and Feature-oriented Product Database for Collaborative Engineering Collaborative Product Design & Manufacturing Methodologies and Applications [M], Edit by Li W D, Ong S K, Nee A Y C and McMahon C A, published by Springer, England, 2007: 109-136
- 3 Dohmen M, Constraint-based feature validation [D], Netherlands, Delft University of Technology, 1998

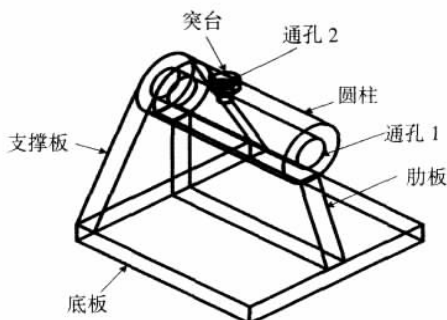


图 4 更新后的实例模型
Fig.4 Modified part