

ICS XXXX

CCS XXXX

T 团 体 标 准

T/SCGS XXXXXX—XXXX

基于语义特征的三维模型部件构建与编辑 技术规范 (征求意见稿)

Technical specification for Construction and Editing of Three-Dimensional
model components based on semantic features

XXXX-XX-XX发布

XXXX-XX-XX实施

中国图学学会 发布

目 次

前言 I

引言 II

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 缩略语 2

5 系统框架 2

6 语义特征的分类与表达规范 3

6.1 语义特征分类体系 3

6.2 全局语义 4

6.3 特征组件 4

6.4 特征关系 6

7 语义特征驱动的部件生成技术规范 7

7.1 生成流程与机制 7

7.2 语义解析与参数提取规范 8

8 编辑 8

8.1 参数修改 8

8.2 特征添加/删除 8

8.3 拓扑变更 9

8.4 撤销与重做 9

9. 存储 9

9.1 模型存储要求 10

9.2 模型存储格式 10

10. 示例 11

10.1. 示例一：基础部件的语义特征建模 11

10.2. 示例二：参数化拉伸与局部网格重构 12

10.3. 示例三：参数化缩放与接触面网格协调 13

10.4. 示例四：组件删除与局部网格修补 14

10.5. 示例五：PMI 与全局/局部语义管理 16

10.6. 示例六：撤销/重做与分支管理 16

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本文件由北京大学提出。

本文件由中国图学学会归口。

本标准主要起草单位包括：北京大学、中国船舶集团海舟系统技术有限公司、北京云境智仿信息技术有限公司、北京微视威信息科技有限公司、清华大学、北京航空航天大学、湖南大学等。

本标准主要起草人包括：朱飞、盖孟、汪国平、张举、姜凯瑞、张孝铖、潘佳骐、许嘉城、赵泽民、王建民、宁涛、王琥、陈彦如、李胜、夏勇峰、朱煜、逢博、汪汝一、张桐源等。

本文件为首次发布。

引 言

随着工业软件的快速演进，实现计算机辅助设计（CAD）与计算机辅助工程（CAE）的无缝衔接已成为复杂产品研发的关键技术瓶颈。当前主流的三维模型数据格式普遍基于边界表示法（B-Rep）及 STEP 等中性数据交换格式，存在高层工程语义缺失、内部拓扑表达不足、离散网格修改困难等突出问题，导致模型在跨平台流转与多学科协同过程中频繁出现信息丢失与数据断层，严重制约了 CAD/CAE 一体化进程。与此同时，生成式人工智能技术在三维设计领域的应用日益受到关注，但现有数据表达方式难以提供具备工程语义、可编辑且符合人类设计逻辑的构建历史信息，使得智能化三维生成技术缺乏可依赖的底层数据结构支撑。因此，亟需建立一种新的三维模型数据表示与编辑规范，从底层数据结构层面突破传统表示方法的固有局限。

本文件旨在规范基于语义特征框架（Semantic Feature Framework, SFF）的三维模型部件构建与编辑技术，通过构建包含全局语义、特征组件与特征关系三个层次的多层次架构，实现高层设计意图与底层几何拓扑的深度解耦与动态关联。本文件规定了语义特征框架的总体架构、数据结构设计、特征编辑操作流程、基于特征历史树的存储与回溯机制，以及多学科统一参数化定义方法，涵盖物理属性、产品制造信息（PMI）及面向 CAE 的仿真约束的统一封装与管理。通过引入基于索引的轻量化映射机制与关系解耦策略，本文件支持意图驱动的智能模型编辑、局部网格自适应重构，以及编辑历史的可追溯与分支管理，为三维模型在工业制造、数字孪生、游戏娱乐等领域的跨平台应用提供统一的技术依据。

本文件的制定填补了国内在语义特征驱动的三维模型构建与编辑技术领域标准规范的空白，为相关软件系统的研发、评测与互操作提供了统一的技术准则。本文件的实施有利于促进产品设计技术的规范化发展，降低多学科协同设计中的数据壁垒与信息损耗，提升复杂产品研发效率；有利于推动三维模型数据结构向语义化方向演进，为生成式人工智能在工业设计领域的深度应用奠定数据结构基础；有利于引导产业各方在统一规范下开展技术创新与产品研发，形成良性的技术生态。

基于语义特征的三维模型部件构建与编辑技术规范

1. 范围

本文件规定了面向产品设计的语义特征驱动三维模型部件构建与编辑的总体架构、语义特征分类与表达规范、部件构建技术规范、部件编辑技术规范以及 CAD/CAE 协同与转换规范等核心内容，明确了语义特征、几何特征、边界特征、面特征及离散特征之间的数据映射与关联机制。

本文件适用于产品设计的工业软件研发、三维几何建模平台建设、仿真分析前处理系统开发以及基于模型的系统工程等相关技术领域。可作为三维模型参数化生成、基于语义的几何与拓扑编辑、CAD 设计模型向 CAE 分析模型自动转换与简化的技术依据。

2. 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

- GB/T 5271.1 信息技术 词汇 第1部分：基本术语
- GB/T 11457 信息技术 软件术语
- GB/T 17304 CAD通用技术规范
- GB/T 28174 三维模型数据要求
- GB/T 36341 形状建模信息表示

3. 术语和定义

GB/T 11457-2006、GB/T 17304-2009界定的以及下列术语和定义适用于本文件。

3.1.

语义特征 semantic feature

附加在三维几何模型之上，能够表达工程意图、编辑要求、物理属性的结构化信息

3.2.

语义特征框架 Semantic Feature Framework

定义在几何体正则表面上的多层次拓扑结构，由全局语义、特征组件与特征关系构成，以轻量索引映射实现高层语义与底层几何拓扑的解耦关联，支持意图驱动的智能编辑与局部网格自适应重构。

3.3.

几何特征 geometric feature

构成三维模型形状的点、曲线、曲面及实体等几何元素的空间位置与解析表达信息。

3.4.

边界特征 boundary feature

描述几何特征之间相互连接关系、邻接状态及包围方向的模型拓扑结构信息。

3.5.

面特征 face feature

构成模型拓扑结构的面单元信息。

3.6.

离散特征 discrete feature

三维模型经过网格划分或离散化处理后，生成的由节点、网格边、网格面组成的离散表达信息。

4. 缩略语

B-Rep: 边界表示法 (Boundary Representation)

V-Rep: 体表示法 (Volumetric Representation)

CAD: 计算机辅助设计 (Computer-Aided Design)

CAE: 计算机辅助工程 (Computer-Aided Engineering)

SFF: 语义特征框架 (Semantic Feature Framework)

SFDG: 语义特征图 (Semantic Feature Dependency Graph)

FHT: 特征历史树 (Feature History Tree)

PMI: 产品制造信息 (Product Manufacturing Information)

DAG: 有向无环图 (Directed Acyclic Graph)

STEP: 产品模型数据交换标准 (Standard for the Exchange of Product Model Data)

5. 系统框架

为实现意图驱动的三维模型智能编辑,提升 CAD 工具的智能性,同时消除 CAD 与 CAE 之间的信息孤岛,本文件提出的 SFF 框架旨在支持高效的模型特征部件编辑与后续仿真分析。

SFF 将复杂的三维模型工程信息抽象为三个核心维度:全局语义 (Global Semantics)、特征组件 (Components) 与特征关系 (Relations)。三者相互协同,共同构成了一个完整的知识表达与管理网络,具体关系如图 1 所示。

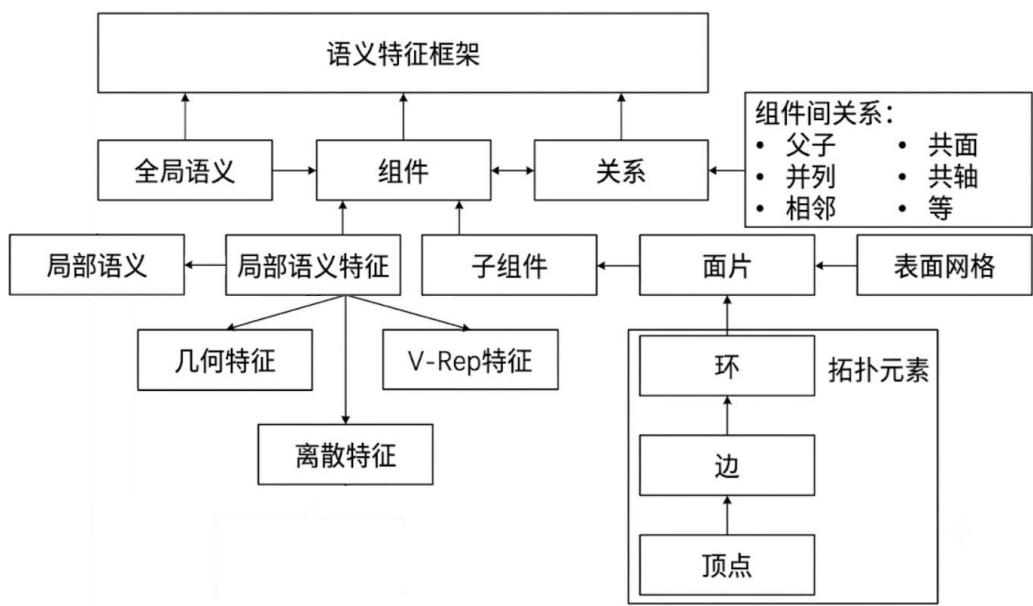


图 1 语义特征框架图

6. 语义特征的分类与表达规范

6.1. 语义特征分类体系

6.1.1. 几何语义

几何语义用于描述和控制三维模型部件的形状、尺寸、空间位置及纯数学解析特征。它是驱动底层几何内核生成点、曲线、曲面及实体解析表达的核心输入。几何语义应包括但不限于：

- a) 基本解析体语义：定义基于标准代数方程的经典解析几何体或解析曲面。包括平面、圆柱面、圆锥面、球面及环面等。该语义通过显式的数学参数直接驱动底层解析方程的生成。
- b) 参数化自由曲面/曲线语义：定义无法用简单代数方程表示的复杂几何元素，例如 Bézier、B样条、NURBS曲线/曲面。该语义应包含控制多项式方程的数学要素，如控制点坐标、控制多边形或者控制网格、权重、节点向量以及曲线/曲面的阶数。
- c) 空间变换与坐标系语义：定义几何元素在三维笛卡尔坐标系中的绝对或相对位置关系。该语义通过齐次坐标变换矩阵来驱动几何体在空间中的位置。

6.1.2. 拓扑语义

拓扑语义用于描述和控制三维模型中各拓扑实体（顶点、边、环、面、壳、体块）之间的连接关系、邻接状态及边界包围方向。它是将独立的解析几何元素“缝合”为完整、封闭且具有明确内部/外部定义的空间结构的核心依据。拓扑语义应包括但不限于：

- a) 边界与包围语义：定义低维拓扑实体如何构成高维拓扑实体的边界。例如顶点对边的界定、由首尾相连的边组成的环对面的裁剪与界定、由闭合的面组成的壳对体块的空间包围关系等。
- b) 邻接与连通语义：定义同维或跨维拓扑实体之间的空间连通状态。例如相邻面之间通过共享边实现的拓扑连续、相邻边之间通过共享顶点实现的连通等。

c) 拓扑持久化语义：定义在模型编辑或特征生成过程中，拓扑元素发生分裂、合并或删除时，其连接关系的继承与追踪规则。确保当几何参数改变导致拓扑结构重建时，原有的边、面连接关系等语义不丢失。

6.1.3. 离散语义

离散语义用于描述和控制连续的几何与拓扑模型向离散网格模型转换的规则与约束。它是驱动网格划分引擎生成离散特征（节点、边、面、体单元）的核心输入。离散语义应包括离散边、离散点等信息。

同时描述几何形体的细分曲线曲面也可以通过离散语义来确定，细分曲线曲面通过细分初始控制多边形或者控制网格，以及相应的细分迭代格式构成细分曲线曲面构成的实体表面网格或者实体内部网格的离散语义。

6.2. 全局语义

SFF 架构采用了一种全局共享与局部自治相结合的混合设计模式，SFF 设立了全局语义 (Global Semantics) 模块，作为统筹管理模型中具有普适性、共享性及跨层级约束性质的宏观工程信息的公共数据列表。

a) 全局产品制造信息(PMI)：PMI 涵盖了零件制造与检验所需的所有关键信息，包括几何尺寸与公差(如长度、直径、圆柱度、平面度等)、表面粗糙度、材料与工艺规范，以及三维注释(如基准标识、文本说明等)。全局语义作为承载 PMI 的核心载体，记录了贯穿产品全生命周期的全局性规范。

b) 共享物理与材料属性库：为便于物理仿真，全局语义内置了结构化的共享材料库，封装了工程常用的材料模型属性。该设计允许模型中大量同质化零部件通过轻量化引用机制直接获取材料数据，从而显著降低数据冗余。

c) 全局仿真环境与边界条件：全局语义负责记录物理仿真中的宏观环境约束，如全局重力场方向、环境基础温度或流场初始参考压力等。这些信息对模型中的所有子部件均具有普遍的约束效力。

6.3. 特征组件

在 SFF 架构中，特征组件 (Components) 是承载具体工程功能、设计意图以及局部语义的实例化核心单元。为了适应复杂多变的 CAD/CAE 协同设计需求，SFF 赋予了特征组件一定的自由度与灵活性。

6.3.1. 语义特征框架

语义特征框架由底层几何体的多条特定边相互连接而形成的闭合线框构成。这种线框结构的生成与几何体的局部区域语义特征密切相关。在三维模型中，几何体的特定局部几何块可被赋予特定的工程语义信息，其本身也是一种拓扑结构。由这些带有语义的区域几何块的边界所构成的空间框架，即构成语义特征框架中的一个独立闭合线框。如果将整个几何体的表面区域划分为多个带有独立语义的表面区域块，这些表面区域块的边界线框相互组合、拼接，便构成一个完整的几何体表面语义特征框架。

SFF 是一个定义在几何体正则表面上的正则拓扑结构，SFF 边界线框是几何体的边界线 (Edge)，这些边界线构成 SFF 的特征区域边界线 (Patch Boundary Line)，SFF 的边界线 PBL 围成的最小内外环 (Loop)，Loop 张成的 SFF 拓扑表面 (Patch)，SFF 的拓扑表面 Patch 可以与几何体的表面不重合。SFF 的边界线框 PBL 和拓扑表面 Patch 构成的集合是正则表面，满足欧拉公式。

语义特征框架利用几何体的拓扑结构表示。语义信息可绑定在语义特征框架的任意元素上，例如：

- a) 框架节点（VertexNode）
- b) 框架边（Edge）
- c) 框架的特征区域边界线（Patch Boundary Line）
- d) 框架的特征区域（Patch）
- e) 框架的特征块（Block）
- f) 特征框架（Wireframe）
- g) 整个拓扑表面（Body）等。

构建语义特征框架的核心目的，在于实现对几何体局部区域的意图驱动与智能编辑，从而实现几何体局部特征的直接建模。框架为定义在区域几何块或区域表面上的各种语义属性信息提供结构化的载体，为几何体的局部编辑提供操作选项。在底层几何体为离散网格体时，操作海量且无序的网格节点极其困难且容易破坏拓扑。而依托于语义特征框架，用户只需对带有语义的闭合线框或特征区域进行操作，系统即可通过底层的映射关系，自动定位并驱动局部网格的自适应修改与重构。这种基于 SFF 的编辑方式比 FFD（Free Form Deformation）编辑操作具有更小的编辑粒度，编辑操作的形状控制效果更好，更符合用户对产品设计意图的编辑要求。

6.3.2. 基于索引的轻量化组合机制

特征组件在组合上述拓扑元素时，采用高度解耦的轻量化逻辑架构。特征组件自身既不存储庞大的几何坐标数据，也不会主动记录底层的拓扑结构。

特征组件作为高层逻辑容器，负责对体、特征框架、特征区域、区域边界线、框架边以及框架节点等中层元素进行灵活组合与管理。体、特征框架、区域、边界线等中层元素与底层 V-Rep 结构或离散网格体产生逻辑绑定，通过直接记录底层几何拓扑结构中顶点、边、面等拓扑结构的索引值（Index）来实现精确映射。

SFF 框架为所有的顶层结构（特征组件）与中层结构（体、框架、区域等）均分配了独立的自身索引值（Self-Index）。依托于这种自身索引机制，特征组件不仅可以组合中层元素，还可以直接通过索引引用其他特征组件，从而实现组件包含组件的深度嵌套结构。

当底层几何发生平移、旋转等不改变拓扑的刚体变换时，高层组件无需进行任何数据更新。而当进行局部网格重构时，只需更新受影响的中层元素内部的底层索引映射关系，顶层的组件嵌套逻辑依然保持稳定。

6.3.3. 局部语义的自由定义与属性覆盖

SFF 在特征组件使用全局引用与局部自治相结合的双重语义解析机制。

特征组件默认采用指针引用的模式。组件内部存储一个或多个指向全局语义的唯一标识符 ID，以此来获取共享的材料库、全局公差标准或宏观环境载荷。当发生全局性的设计变更时，只需在全局语义中修改一次，所有通过 ID 引用该属性的特征组件均会实现状态的实时同步。

当某个特征组件具有特殊的工程需求时（异质材料建模、局部计算与编辑处理等复杂场景），系统允许在该组件内部直接创建并定义其专属的局部语义。

SFF 在底层建立一套严格的属性寻址与优先级解析规则。当系统或下游查询某个特征组件的特定属性，系统首先检索该组件内部的局部语义记录。如果存在局部定义，则优先直接读取并返回该值，此时全局语义中的同类属性应被自动覆盖。当局部语义中未定义该属性时，

系统会根据 ID 指针，向上回溯并读取全局语义中的默认值。

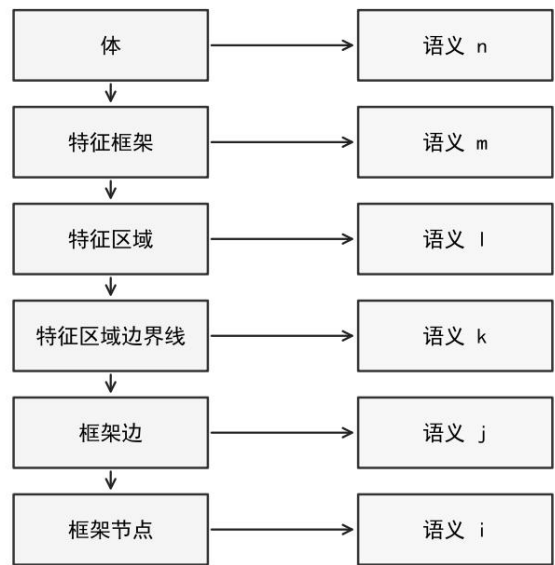


图 2 几何体的 V-Rep 结构图

6.4. 特征关系

在 SFF 架构中，特征关系（Relations）被抽象为一类独立的核心数据对象，专门用于精确定义、记录并维护特征组件（Components）之间的多维依赖联系。

6.4.1. 语义特征依赖图

特征关系（Relations）对象的核心在于特征组件（Components）自身支持深度的层级嵌套机制。

当跨层级、跨分支的横向特征关系与纵向的组件嵌套逻辑组合在一起时，多个组件之间便形成了一个复杂的网络结构，这种结构称为语义特征依赖图（Semantic Feature Dependency Graph，简称 SFDG）。图 3 展示了上段中所描述的结构 SFDG 图。

SFDG 是一种典型的有向图。这个图中有如下结构：

- a) 节点（Nodes）：代表各个层级的特征组件。
- b) 有向边（Directed Edges）：代表特征关系。边上附着的属性则记录了关系的具体类型与约束参数。

SFF 框架通过将特征关系抽象为 SFDG 中的独立边对象，将实体组件与链接关系进行解耦。组件 A 和组件 B 之间不再直接互相存储对方的内存地址。相反，它们都只与一个独立的特征关系对象 R 进行联系。关系对象 R 作为中间层，专门负责记录组件 A 和组件 B 之间的约束关系。

6.4.2. 特征关系的分类体系

SFF 框架支持以下四大类特征关系的定义与维护：

- a) 拓扑与空间装配关系：构建复杂三维模型的基础约束形式，用于界定特征组件在三维空间中的相对位置、姿态以及严格的几何边界约束。常见的特征组件之间的拓扑约束关系包括但不限于：邻接关系、父子关系等层次关系、并列关系等各种拓扑关系，空间装配关系包括但不限于：共面、共轴、平行等约束关系。

- b) 逻辑与参数依赖关系：用于记录基于工程规则的数学驱动意图与逻辑层级。它通常表现为参数间的代数方程、条件判断或父子层级依赖。当核心设计参数发生变更时，系统能够沿着逻辑关系链，自动推导并更新所有相关组件的几何形态。
- c) 物理关系：用于协助物理仿真。明确界定这些场之间的交互界面。通过在 SFF 中显式地定义物理关系，下游的 CAE 求解器可以直接读取并映射相应的边界条件与载荷传递矩阵，实现设计模型与分析模型的无缝数据衔接。

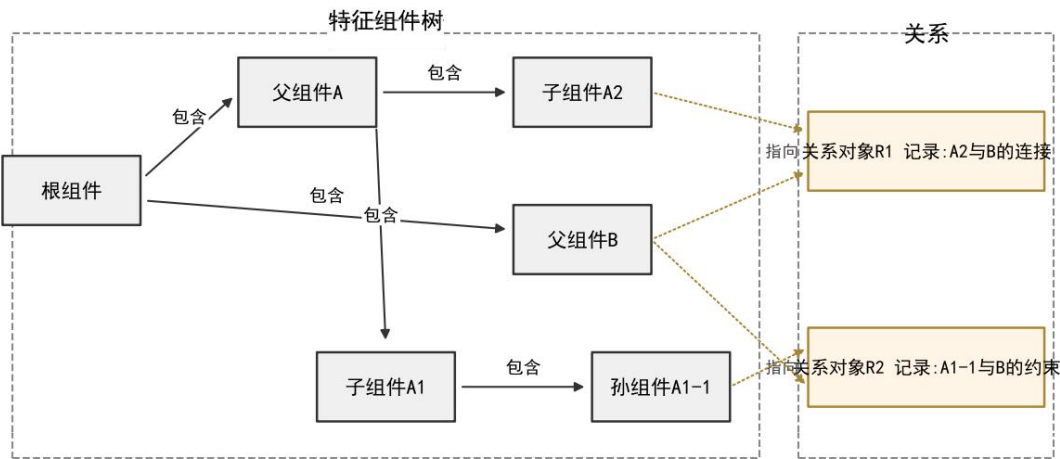


图 3 语义特征依赖图示例

7. 语义特征驱动的部件构建

7.1. 部件构建流程与机制

7.1.1. 标准的部件构建流程

部件的构建应严格遵循以下单向数据流：语义输入与校验->语义解析与参数提取->几何特征实例化->拓扑重建与布尔运算->面特征生成与语义绑定->离散特征构建->离散特征语义绑定。

这里的部件构建，可以从某个部件表面上形成一个二维图元再通过参数做拉伸等基本操作手段形成一个三维几何特征，也可以是另一个特征组件的拷贝移植、也可以通过把部件的形变操作等手段构建，这些过程都会造成拓扑结构的改变和更新，SFF 也会更新。

7.1.2. 多几何内核调度

在多几何内核兼容的架构下，系统应建立统一的“语义-内核”适配层。特征组件构建机制应将提取的通用语义参数，动态翻译为当前激活的不同集合内核的特定 API 调用指令，确保同一组语义在不同内核下生成的几何与拓扑结果保持一致。

7.2. 语义解析与参数提取规范

本节规定了如何从结构化的语义数据中提取用于驱动底层数学计算的有效参数。

7.2.1. 几何参数提取与转换

系统应能从几何语义中准确提取显性几何参数如长、宽、半径等。对于参数化自由曲面，应能提取控制点坐标及权重向量，并将其转换为底层内核所需的标准数学格式（如 NURBS 定义等）。

7.2.2. 语义合法性校验

在参数提取阶段，系统必须进行工程规则与边界条件的越界检查。例如，校验孔径参数是否大于所在面的宽度，或抽壳厚度是否导致自交。若校验不通过，系统应中止构建流程并抛出异常代码。

8. 编辑

本章规定了基于语义特征框架的四类特征编辑操作：参数修改、特征添加/删除、拓扑变更以及撤销与重做。所有编辑操作均应遵循语义优先、局部更新的原则：系统首先解析编辑指令中的语义参数，通过语义特征识别受影响的组件与关系范围，然后仅对受影响区域执行拓扑更新与局部网格重构，最后将编辑前后的状态差异以增量变更形式记录至特征历史树，并更新整个模型的拓扑结构以及 SFF。

各类编辑操作在执行前应进行前置条件检查，包括目标组件存在性验证、参数有效性验证、约束冲突检测等。在执行过程中应保持特征关系的拓扑有效性，对因编辑操作而失效的关系进行标记或修复。在执行完成后应触发约束传播，确保模型全局一致性。

8.1. 参数修改

本节规定了通过修改语义特征的属性参数来驱动模型更新的规范。

8.1.1. 几何参数驱动

修改几何语义中的几何参数时，系统应触发底层解析几何方程的重新计算，并驱动相关拓扑边界的自动更新。

8.1.2. 级联更新

系统应根据特征树中的特征组件的父子依赖关系，按顺序对受影响的下流特征进行级联更新，确保整体模型的数学解析与拓扑连接保持一致。

8.1.3. 离散语义更新

当修改离散语义时，系统不应重构底层几何与拓扑，而应根据特征组件的几何参数直接驱动网格划分引擎重新生成离散特征。

8.2. 特征添加/删除

8.2.1. 基于框架的特征实例化与添加

添加新特征时，必须以语义特征框架为基准进行实例化，并生成新的相应的语义节点。新添加的的语义节点插入到框架的特征树中，并明确其父级依赖关系。

8.2.2. 框架内的依赖性检查与特征删除

特征删除操作在本质上是语义特征框架中特定语义节点的移除。在执行删除前，框架必须基于依赖树进行依赖性检查。若被删除的特征节点存在下游子节点，框架向用户或上层应用抛出框架级警告。

8.2.3. 语义属性的继承与剥离

当在框架内删除某一结构特征时，附着在该特征载体上的属性信息、语义信息必须由框架统一执行同步剥离或清理。

8.3. 拓扑变更

本节规定了因参数修改或特征增删导致模型底层拓扑结构（面、边、顶点的数量或连接关系）发生本质改变时的处理规范。

8.3.1. 拓扑持久化

系统应建立稳健的拓扑命名与追踪机制。当拓扑变更导致一个面分裂为多个面，或多个面合并为一个面时，系统应能追踪其演变历史，确保原有的拓扑标识不丢失。

8.3.2. 属性重映射

发生拓扑分裂时，原附着于母面上的属性信息应按相应规则映射至分裂后的所有有效子面上。发生拓扑合并时，应进行语义冲突检测与合并。

8.3.3. 离散特征的局部重构

拓扑变更发生后，系统宜采用基于几何特征的局部网格重划分技术，仅对拓扑发生改变的局部区域进行离散特征更新，以提高计算效率。

8.4. 撤销与重做

本节规定了系统状态回溯与恢复的机制。

8.4.1. 语义级状态管理

撤销与重做操作应建立在“语义特征级”而非“底层几何级”。状态栈中记录的应是语义参数的变更指令、特征的增删动作及拓扑语义的修改记录。

8.4.2. 多数据层同步

执行撤销/重做时，系统应保证几何语义、拓扑语义和离散语义等语义信息的严格同步。不允许出现部分语义撤销但是另一部分语义未修改的状态。

8.4.3. 保护机制

撤销/重做机制应具备保护机制。若某一步回滚计算失败，系统应能整体回退至上一个安全状态，确保模型不被破坏。

9. 存储

本章规定了语义特征模型在持久化存储过程中的核心要求与格式规范。首先，从数据完备性、拓扑语义映射及属性可扩展性三个维度，明确模型存储应满足的基础条件，确保模型

在跨软件、跨几何内核环境下重构时，几何、拓扑与属性信息能够完整保留并准确还原；其次，提出基于语义特征框架节点的层级存储架构，阐述数据与属性相分离的存储机制，并定义支持大型复杂装配体分布式协同设计的跨文件引用机制——包括局部命名空间隔离与非破坏性属性覆盖，以保障多学科、多专业团队并行工作场景下的数据一致性、安全性与可扩展性。

9.1. 模型存储要求

9.1.1. 数据完备性要求

模型存储文件应完整包含部件的特征依赖树、各层级的语义特征参数（几何、拓扑、离散语义）、底层几何内核生成的边界表示（B-rep）数据，以及相应的属性数据标签。

9.1.2. 拓扑语义映射

存储模型必须包含语义标签与底层拓扑实体之间的映射表。确保模型在不同软件或不同几何内核中重新加载并重构时，属性、边界条件等信息能够准确找回其依附的拓扑面或边。

9.1.3. 属性扩展要求

存储结构应具备高度的可扩展性，允许上层应用在不破坏原有几何与拓扑数据结构的前提下，动态追加自定义的属性。

9.2. 模型存储格式

9.2.1. 基于框架节点的层级存储架构

模型的存储结构应表现为一棵由语义特征框架节点构成的树。每个节点代表一个独立的语义特征框架。节点之间通过父子关系或引用关系进行组织，完整记录模型从宏观装配到微观几何特征的生成逻辑与依赖链条。

9.2.2. 数据与属性分离的存储机制

节点内仅存储的用于生成该特征所需的最小化几何参数、底层内核生成的轻量化边界表示（B-rep）缓存，以及用于跨节点引用的拓扑标识符，应与特征之上的属性信息分离。

9.2.3. 跨文件引用

模型存储格式应具备外部文件引用机制，支持将大型复杂装配体拆解为多个子文件进行分布式存储，以满足多学科、多专业团队的并行协同设计需求。当前语义特征框架内的节点可直接指向并引用外部独立的语义特征描述文件；在解析时，被引用的外部文件将作为完整的逻辑子树，动态挂载并实例化于当前节点的层级结构之下。

为保障数据的一致性与安全性，系统在引入外部引用时，必须为被挂载的语义特征子树自动分配独立的局部命名空间，从而彻底杜绝跨文件合并与装配过程中可能产生的拓扑标识符或特征名称冲突。

此外，存储架构应支持基于层级的“非破坏性属性覆盖”机制。允许上层框架节点对所引用的底层外部节点属性进行重定义或修改；该修改仅在当前文件中以“覆盖指令”的形式记录，绝不篡改或破坏原始被引用的外部源文件。

模型存储时的外部文件引用关系，通过模型根文件夹中的 json 说明文件记录各种引用与特征文件依赖关系，在 json 文件中记载根文件与各个外部特征文件之间的层次引用关

系，同时在 json 文件中记录各个外部文件之间引用关系的相对存储路径，构成一颗外部特征层次引用树结构。

10. 示例

本章通过若干典型应用示例，展示本文件所规定的工程实践。示例中以简化的伪代码与流程描述说明所涉及的核心机制。示例的目的在于帮助理解本文件第 5 章至第 9 章规定的语义特征分类、部件生成、部件编辑及数据存储的相关约定。

10.1. 示例一：基础部件的语义特征建模

本示例展示如何基于 SFF 对一个基础平面加垂直圆柱体装配部件进行语义特征建模，包括特征组件的实例化、语义特征的提取、特征关系的注册以及全局语义的引用。初始状态及语义信息组织如下：

a) 特征组件实例化：

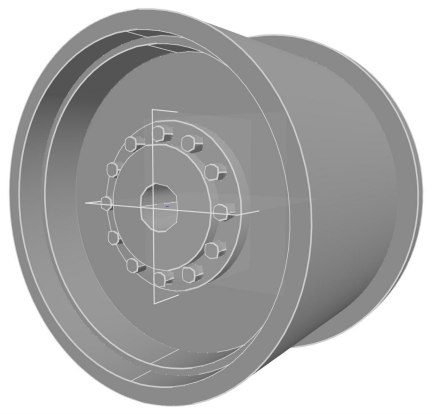
1) Comp_Base（基础平面）：底层绑定 V-Rep 拓扑结构（包含体块 Block 与表面 Patch），并已完成初始表面网格与体网格离散化；

2) Comp_Cyl（圆柱体）：底层同样绑定 V-Rep 拓扑结构；

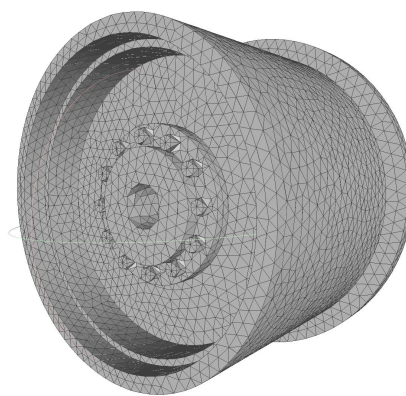
b) 局部语义定义：Comp_Cyl 的 localSemantics 定义为{"length": L0, "radius": R0}；

c) 全局语义引用：Comp_Base 与 Comp_Cyl 均通过 globalSemanticIDs 引用全局共享材料库 Mat_45Steel；

d) 特征关系注册：创建关系对象 Rel_Mate，其 ConstraintProfile 的 type="CO_PLANAR"，parameters={"face_A": "Cyl.bottom", "face_B": "Base.top"}，description="圆柱体底面与基础平面顶面共面"。将该关系注册至 Comp_Cyl 与 Comp_Base 的 relationRegistry 中。



(a) 导入原模型



(b) 原模型网格结构



图 4 基础部件特征组件构建与索引保存示意图

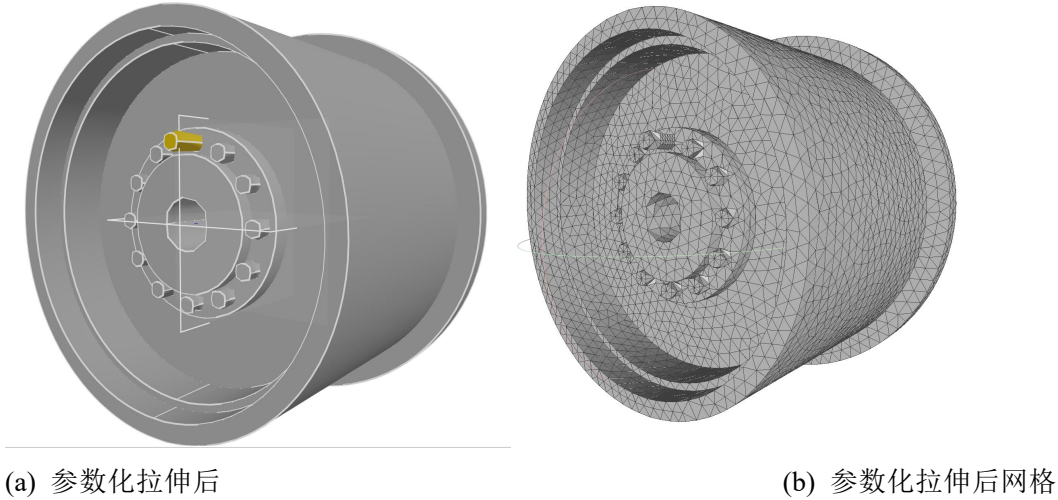
本图展示从原始模型、网格结构到组件提取和组件保存格式的过程，对应“Comp_Base、Comp_Cyl 实例化”“topoElements 索引绑定”“全局语义引用”等。

10.2. 示例二：参数化拉伸与局部网格重构

本示例展示在 SFF 支持下对圆柱体进行参数化拉伸的完整工作流。对圆柱体沿其轴向进行长度从 L0 到 L1 的拉伸（拉伸增量 $\Delta L = L1 - L0 > 0$ ），其处理流程如下：

- a) 步骤一：组件实例化与参数初始化。系统根据设计师要求实例化 Comp_Base 与 Comp_Cyl，并接收用户输入的拉伸参数 L1 与拉伸方向向量 v_dir；
- b) 步骤二：语义参数修改与拓扑更新。Comp_Cyl 的 length 参数由 L0 更新为 L1；底层 V-Rep 结构开始拓扑更新——圆柱体原有侧面沿 v_dir 被延展，并在原底面与基础平面之间生成一个高度为 ΔL 的新几何空腔（NewCavityVolume）；
- c) 步骤三：局部离散网格生成。系统仅针对新拉伸出的 ΔL 空腔区域进行网格划分——沿延展出的新侧面边界自动生成表面网格，封闭该空腔；
- d) 步骤四：SFF 索引映射动态更新。系统将新生成网格的索引（Index_new）追加至 Comp_Cyl 的 topoElements 列表中，并通过 updateMapping 方法完成索引同步；原模型其余离散网格保持不变。

本示例验证了：SFF 使得高层语义参数的修改能够自动驱动底层几何与拓扑的更新，并通过局部重构策略显著降低网格生成代价。



结构

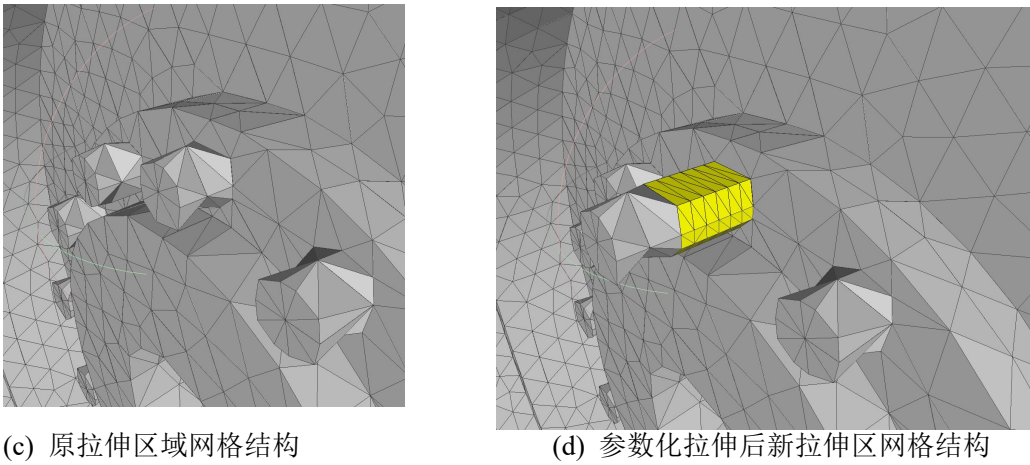


图 5 参数化拉伸中的局部网格重构示意图

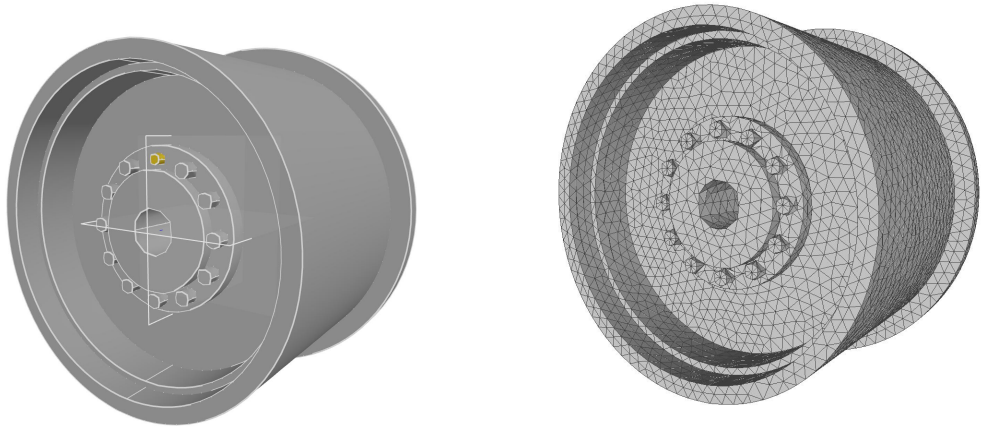
本图用于说明语义参数修改后，系统如何只对新增拉伸区域进行局部网格生成，而不是重构整个模型。对应指标中的“语义参数修改、拓扑更新、局部网格生成、索引映射更新”几个步骤。

10.3. 示例三：参数化缩放与接触面网格协调

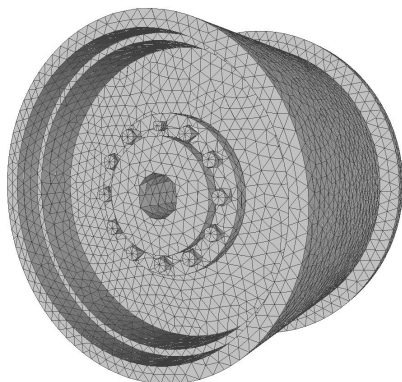
本示例展示在 SFF 支持下对圆柱体进行等比例缩放的工作流。对 `Comp_Cyl` 应用缩放比例 $S>0$ （放大 $S=1.3$ 、缩小 $S=0.7$ 或对孔洞缩放等典型场景），其处理流程如下：

- a) 步骤一：参数初始化。接收缩放比例 S ；
- b) 步骤二：语义参数修改与拓扑更新。`Comp_Cyl` 的 `scale` 参数更新为 S ；底层 `V-Rep` 结构按比例重新计算圆柱体侧面、顶面与底面交界边的几何表达式及顶点坐标；
- c) 步骤三：组件整体离散化与网格生成。系统保留缩放后交界处的离散点位置（已按 S 计算的新坐标），并以这些点为约束点，重新生成组件的网格；
- d) 步骤四：SFF 索引映射动态更新。系统分别调用 `Comp_Cyl` 与 `Comp_Base` 的 `updateMapping` 方法，将圆柱体全新的网格索引以及基础平面上受影响区域的新网格索引同步更新至各自的 `topoElements` 列表中。

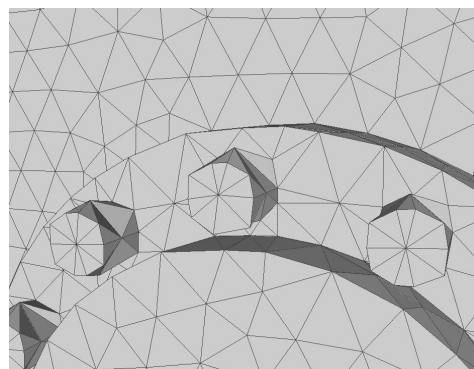
对孔洞组件进行缩放时，由于其约束通常位于前后两个面，孔洞缩放将按两个面的约束进行网格生成与拓扑编辑。本示例验证了：SFF 能够基于关系约束协调多个组件的网格同步更新。



(a) 缩放前网格

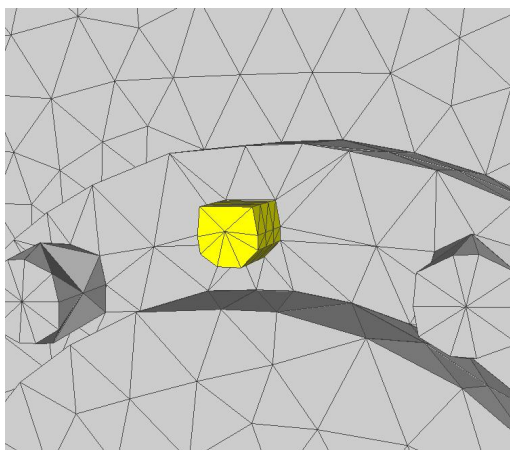


(b) 缩放后结果



(c) 缩放后网格

(d) 缩放前局部区域网格结构



(e) 缩放后局部区域网格结构

图 6 参数化缩放中的几何更新与局部网格协调示意图

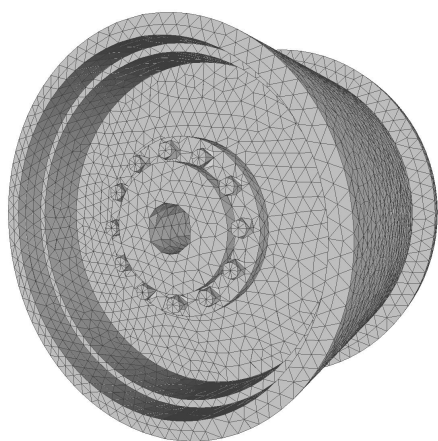
本图说明缩放操作不仅改变组件几何尺寸,还会影响接触区域和局部网格结构。它与“整体离散化、网格重新生成、关系同步更新”的要求对应。

10.4. 示例四：组件删除与局部网格修补

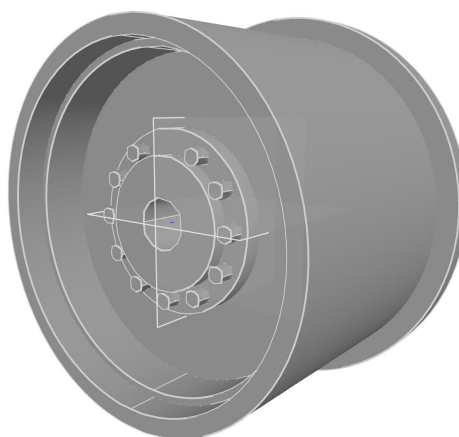
本示例展示在 SFF 支持下删除 Comp_Cyl 的工作流：

- a) 步骤一：触发删除指令与状态初始化。系统接收到针对 Comp_Cyl 的删除指令，定位该组件在 SFDG 中的节点；
- b) 步骤二：组件移除与拓扑分离。系统正式对 Comp_Cyl 执行删除，解除圆柱体底面与基础平面之间的拓扑绑定；基础平面上将留下一个网格空洞 Hole_base；
- c) 步骤三：基础平面的局部网格修复。系统以空洞边界上的原有离散点为约束条件，在空洞区域重新生成网格，从而将原位置的孔洞封闭；
- d) 步骤四：SFF 索引映射动态更新。系统将新填补区域的网格索引同步至 Comp_Base 的 topoElements 中。

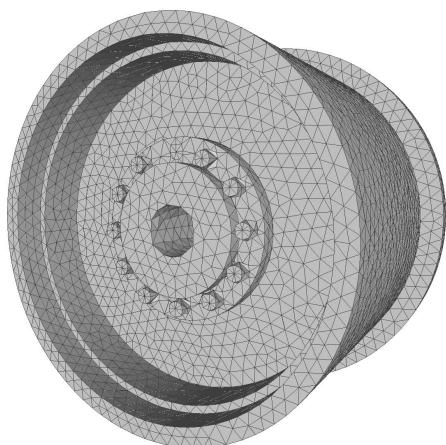
本示例验证了：SFF 能够自动识别被删除组件的影响范围，并在不重构整体网格的前提下完成局部网格修补。



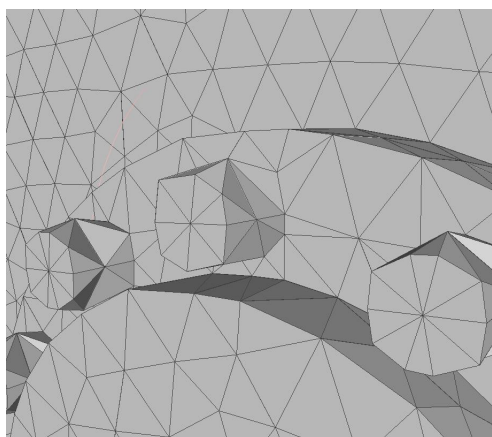
(a) 组件删除前网格结构



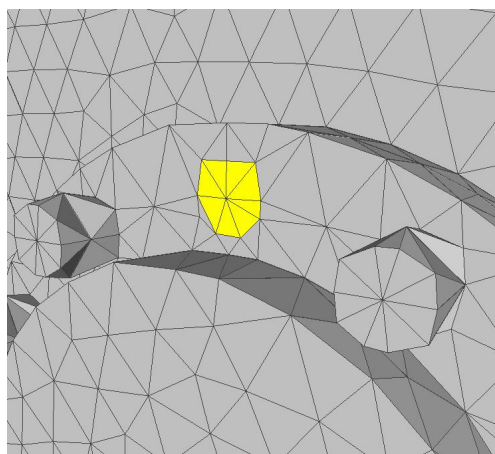
(b) 组件删除后几何结构



(c) 组件删除后网格结构



(d) 组件删除前局部区域网格结构



(e) 组件删除后局部区域网格结构

图 7 组件删除后的局部网格修补示意图

本图可解释删除组件不是简单移除对象，还包括关系清理、拓扑分离、孔洞修补和索引重建。图中黄色区域展示了删除后自动生成的局部网格。

10.5. 示例五：PMI 与全局/局部语义管理

本示例展示在 SFF 中进行 PMI 标注的两种典型模式：

a) 全局 PMI 模式：所有 Comp_Cyl 实例均共享圆柱体直径公差 $\pm 0.05\text{mm}$ 。该公差被定义为全局语义（PMI_Global_Cyl_Dia），Comp_Cyl 通过 globalSemanticIDs 引用该全局 PMI；

b) 局部覆盖模式：特定 Comp_Cyl 实例因装配要求需将公差收紧为 $\pm 0.02\text{mm}$ 。该实例在 localSemantics 中定义{"PMI.dia_tolerance": " $\pm 0.02\text{mm}$ "}, 通过 getSemanticValue 方法自动覆盖全局 PMI 的引用值；

c) PMI 变更的可追溯性：公差值的变更作为 ComponentDelta（path="localSemantics.PMI.dia_tolerance"）记录至 FHT 中，可随时通过撤销/重做操作回溯。

本示例验证了：SFF 能够通过全局引用与局部覆盖的混合机制，在保证宏观 PMI 规范一致性的同时，灵活支持局部特例的工程需求，且全部 PMI 变更均可追溯。

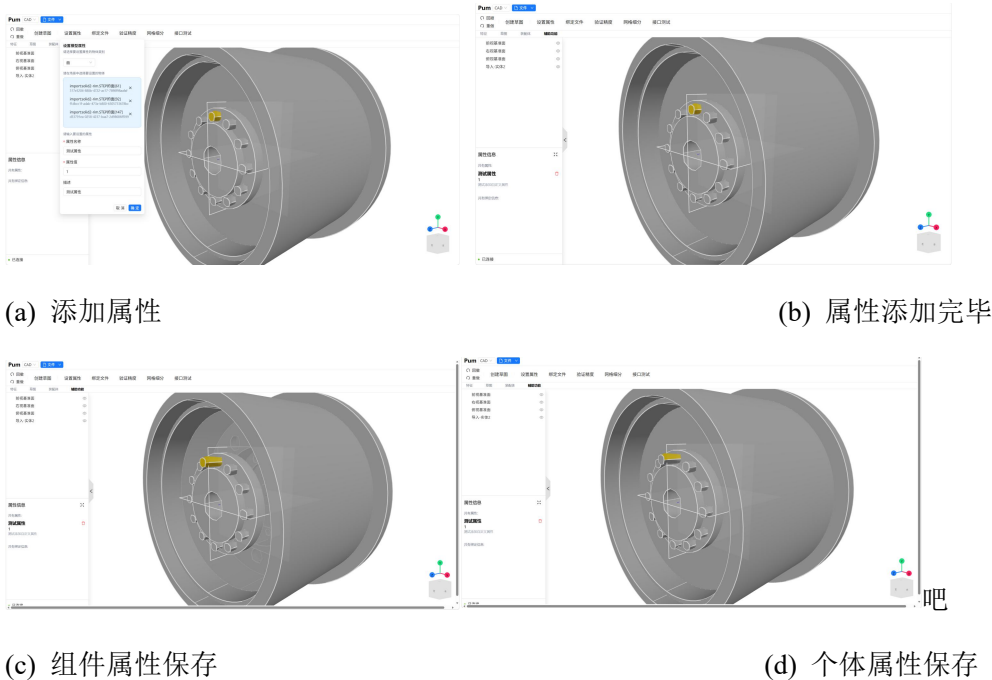


图 8 语义属性独立编辑与稳定继承流程图

本图用于说明属性编辑与几何拓扑编辑之间的解耦关系，对应“属性编辑不应触发几何拓扑变更”和“全局语义、局部语义应保持一致性传播”的要求。

10.6. 示例六：撤销/重做与分支管理

本示例展示 FHT 对非线性编辑流程的支持：

a) 主干编辑：设计师依次对 Comp_Cyl 执行三次编辑操作 Op1、Op2、Op3，FHT 主干路径为 $N0 \rightarrow N1 \rightarrow N2 \rightarrow N3$ ；

b) 分支创建：在 N1 处设计师尝试两种不同方案——方案 A 沿 v_dir 拉伸 (Op2A)，方案 B 沿径向缩放 (Op2B)，系统不覆盖 N2，而是将 N1 标记为分叉点，并创建 N2A、N3A 与 N2B、N3B 两条分支；

c) 分支切换：设计师从方案 A 切换至方案 B 时，系统首先计算两分支的最近公共祖先 N1，然后依次应用逆向变更算子从 N3A 回溯至 N1，再沿分支 B 正向应用 Op2B、

Op3B 得到最终状态；

d) 设计意图保护：方案 A 的所有状态在 FHT 中完整保留，即使后续在方案 B 中进行了激进的拓扑重构，方案 A 依然可以被恢复。

本示例验证了：FHT 的 DAG 结构与分支管理机制能够同时支持高效的撤销/重做与并行的设计方案探索，且全部设计变更均可追溯。

```
{
  "version": 5,
  "componentDelta": { "id": "comp_01", "path":
"localSemantics.thickness", "old": 5.0, "new": 8.0 },
  "relationDelta": { "id": "rel_02", "op":
"UPDATE_CONSTRAINT", "payload": { "distance": 10.0 } }
}
```

图 9 特征历史记录的序列化示例图

参 考 文 献

- [1] GB/T 11457—2006 信息技术 软件工程术语
- [2] GB/T 17304—2009 CAD 通用技术规范
- [3] GB/T 5271.1 信息技术 词汇 第1部分: 基本术语
- [4] GB/T 20000.1 标准化工作指南 第1部分: 标准化和相关活动的通用词汇