

ZW³SOFT

基于特征时序的参数化建模概论

主讲人：黄戈莹

自我介绍

参数化驱动组——黄戈莹

自我介绍

姓名：黄戈莹

籍贯：广东汕头

教育背景：北京航空航天大学 机械工程及自动化专业

工作经历

2022.6年毕业后加入中望，一直在参数化组工作

- 模型数据转化（Creo图纸带参导入ZW3D）
- 轮廓链（拉伸输入中间结构，提升改参稳定性）、HQR
- 历史链数据标准化（非模型数据的其他历史数据管理问题）、属性
- FM重构项目
- 其他日常任务：重生成、特征-实体关系、草图、线框、PMI、特征文件夹etc

C O N T E N T S



三维模型的相关概念介绍

- CAD介绍
- 三维模型表达
- 参数化特征模型相关概念



参数化功能模块

- 特征流程
- 模板命令/特征参数
- Brep数据管理
- 永久命名
- 其他



特征操作（不同时期）实现介绍

- 正向建模
- 回滚
- 播放
- 重定义
- 重生成



参考文档



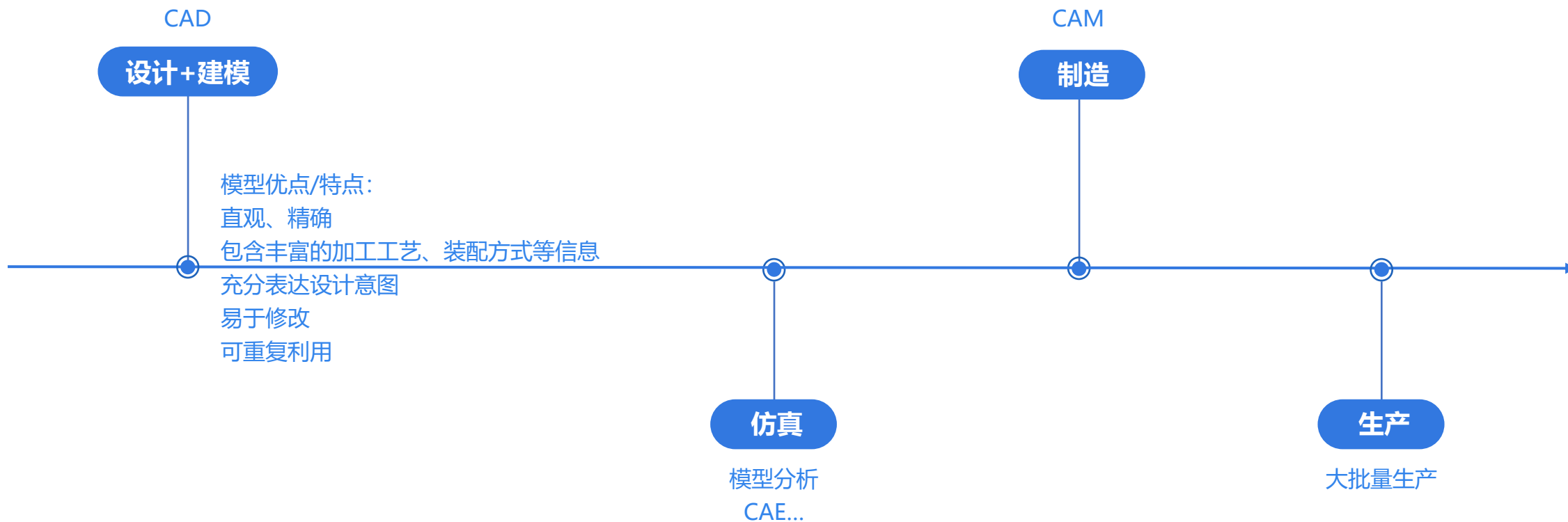
三维模型的相关概念介绍

- CAD介绍
- 三维模型表达
- 参数化特征模型相关概念

CAD介绍

Computer Aided Design 计算机辅助设计

工业生产一般流程：数字化、信息化



三维模型表达

几何模型、参数化特征模型

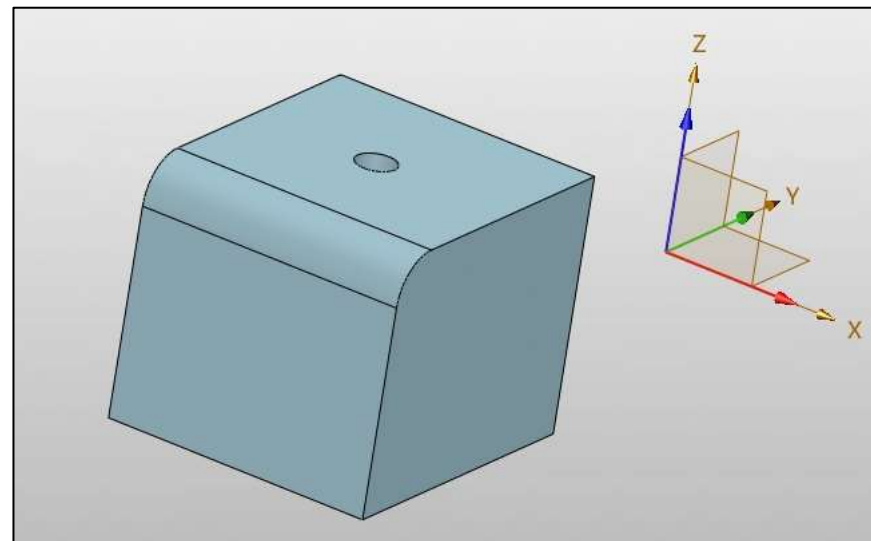
几何模型

边界表示法 (Boundary representation, 简称B-rep/Brep)

根据点, 线, 面这些几何元素构成的表面来对三维模型实体的**几何信息**进行描述, 用顶点, 边, 面, 有向边, 环, 壳, 体等拓扑元素对模型的**拓扑信息**进行描述。

模型结果

- 包含信息少, 没有设计意图及其他生产需要的信息
- 无法使用这个表示方式进行设计和建模



三维模型表达

几何模型、参数化特征模型

特征模型

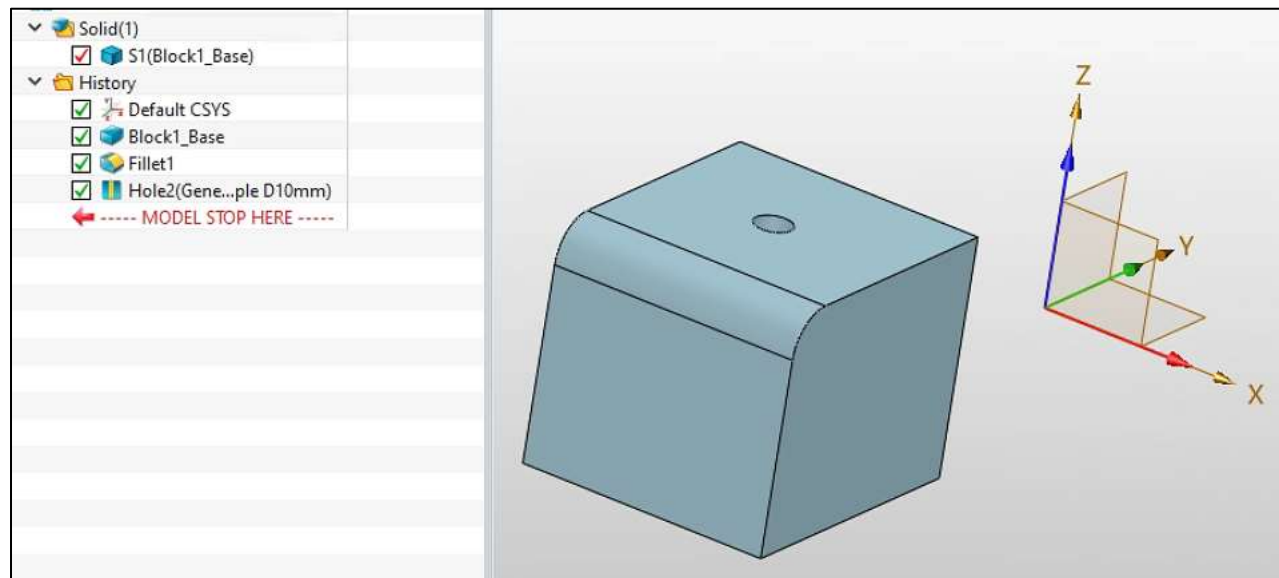
特征

描述一：零件或产品在设计、制造中抽象出来的属性、功能、关系和数据的集合

描述二：具有特定参数和属性并且相互关联的几何形体，是形状、制造加工工艺信息、尺寸等信息的综合描述

模型设计的最小单元

使用“在指定边界生成一个特定尺寸的圆角”、
“在指定面上打一个指定形状的孔”这样的命令
来建立三维 CAD 模型



三维模型表达

几何模型、参数化特征模型

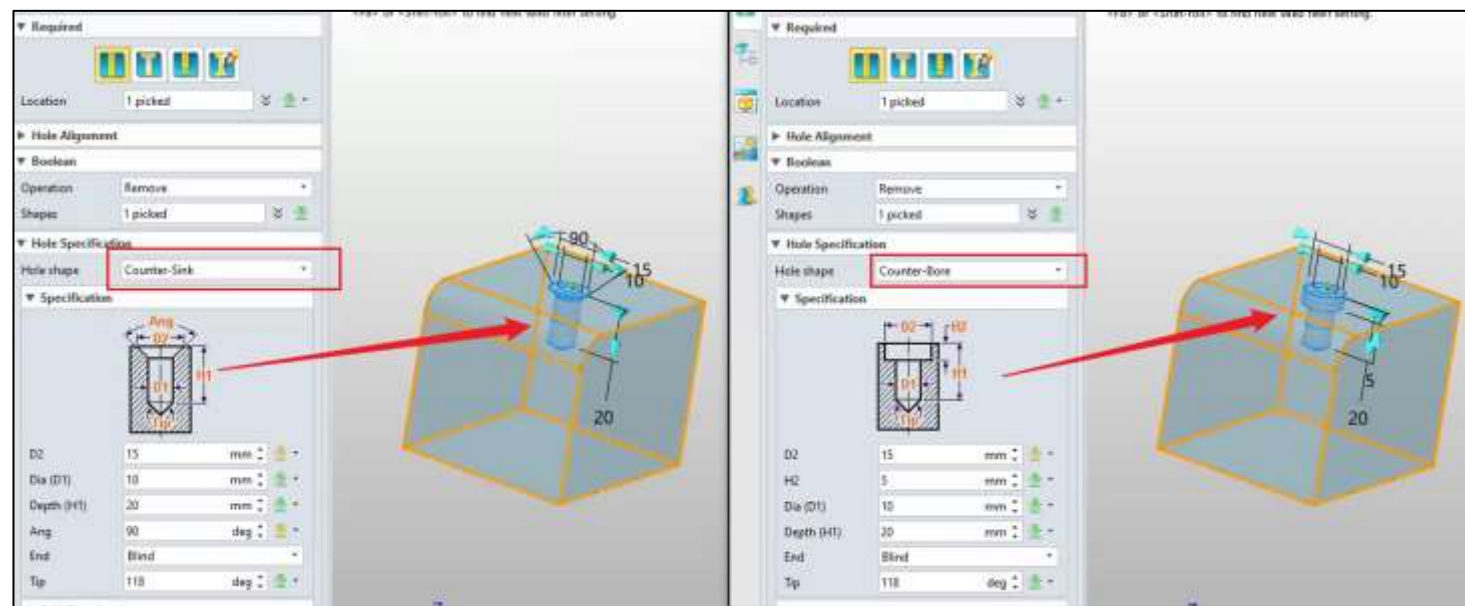
参数化特征模型

参数化建模

建模过程

基于隐含的拓扑约束和工程语义，采用**一组参数**确定产品或零件的几何外形

- 便于修改/模型复用：修改参数就可以驱动生成新的目标几何图形或零件
- 充分表达设计意图：包含了工程语义、加工精度等一系列信息



三维模型表达

几何模型、参数化特征模型

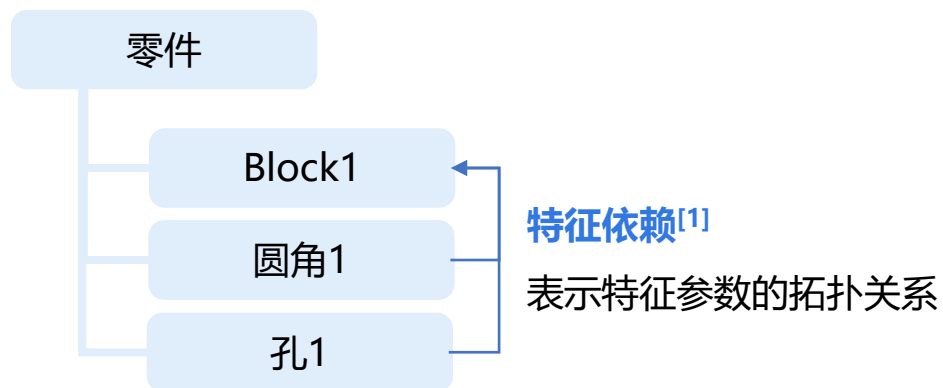
模型

参数化特征模型

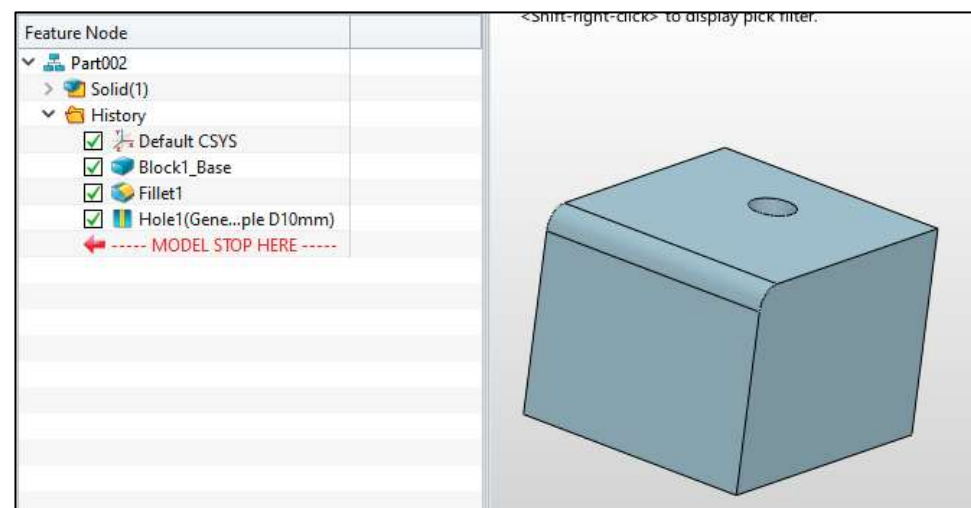
求解

几何模型

特征树 / 历史树



历史链：基于时间顺序，以特征的形式记录参数化建模步骤，所串起的链表结构，以记录用户的参数化设计过程



参数化特征模型相关概念

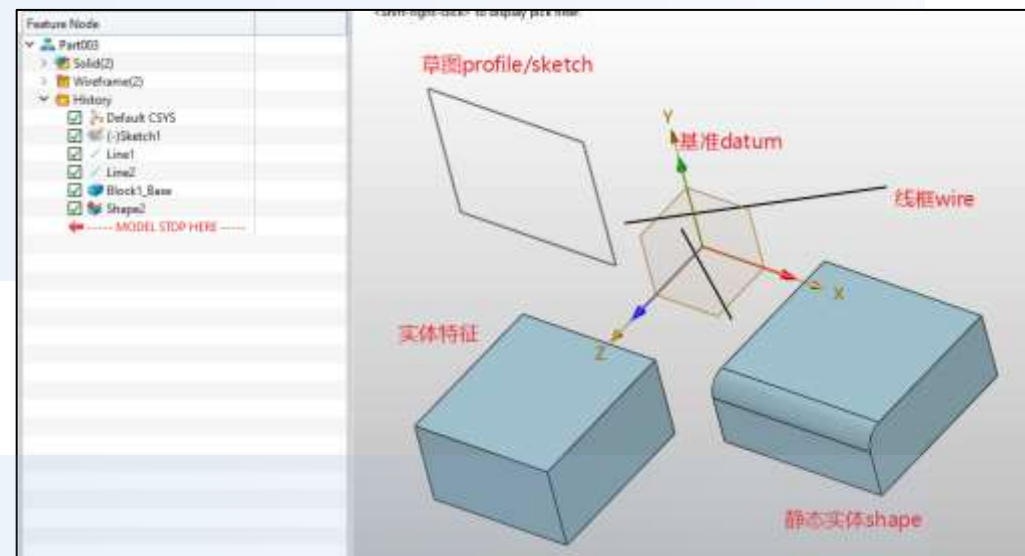
历史树对象

历史树对象：实体特征、草图、基准特征、线框特征、静态特征

实体特征：驱动参数（用户输入参数）和计算结果（几何模型）分离

草图、基准、线框：用于辅助建模，（当前）同时包含驱动参数和计算结果

静态特征：没有历史的，非参数化的，只保留了几何模型信息



特征：普通特征、复合特征

普通特征：包括基础建模特征、草图、基准、线框、静态

复合特征：输入是特征，对特征进行一些操作的特征，包括阵列/镜像和custom特征

custom特征^[2]：基于特征的自定义二次封装，有时候复合特征也单指custom特征，没有很明确的界定

比如刻字命令，是text+拉伸减的组合特征
比如UDF，是用户自定义的特征集合

参数化特征模型相关概念

历史相关操作

正向建模/特征创建create

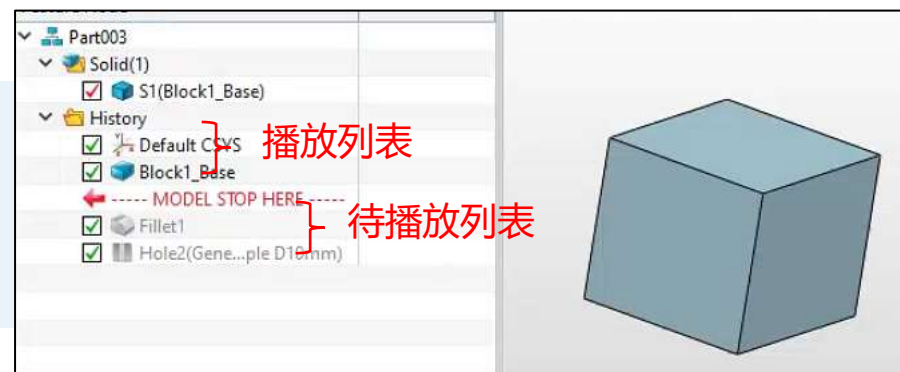
用户输入参数 -> 预览执行结果 -> 确定 -> 执行命令 -> 显示模型结果



回滚rollback

建模条向上移动，模型状态回到第n个特征执行前的模型状态

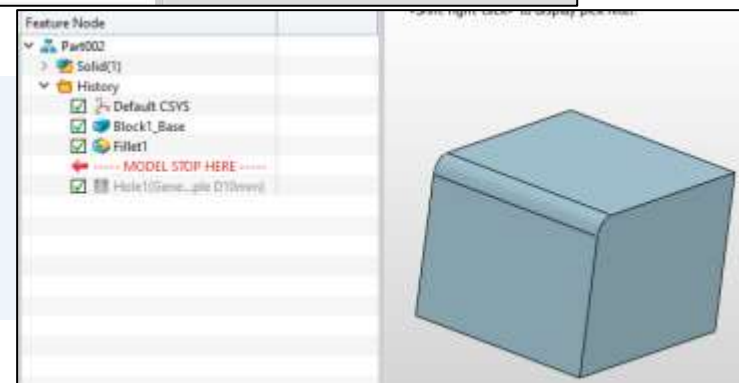
用户触发方式：拖动建模条往上、重定义、抑制/反抑制、删除



播放play

建模条向下移动，显示出后续特征执行结果

用户触发方式：拖动建模条往下、重定义、抑制/反抑制、删除



参数化特征模型相关概念

特征相关操作

特征重定义redefine

目的：通过修改特征参数实现修改模型对应部分几何

特征重生成regen

按照特征中记录的用户输入重新执行这个特征

用户触发方式：删除特征、抑制/反抑制特征、重定义特征，主动拖动建模条播放特征，主动全部重生成，回滚时快速回滚失败，修改标注，删除静态对象（3000之前）

参数化特征模型相关概念

特征相关操作

特征抑制suppress/反抑制unsuppress、删除delete、移动reorder

对历史树上单个特征进行的其他操作，达到修改模型结果的目的

Unlink

改变特征依赖的一个方式

特征依赖

描述：表示特征参数的拓扑关系，来源于VDATA，在特征参数输入确定的时候就已经确定，是一张有向无环图

应用：特征删除、Unlink、抑制、去参、实体特征提取、选择性重生成、循环参考引用、阵列特征成组等



参数化功能模块

- 特征流程
- 模板命令/特征参数
- Brep数据管理
- 永久命名
- 其他

特征流程——历史操作

特征执行流程

历史操作流程

CdCmdHist: 大部分对历史树节点的操作的命令入口

```
int CdCmdHist
(
    const char *s
)
{
    /* ... */
    {
        C_AUTO_LOCK_PMI_REGEN(1);
        bool fSectUpdate = V_FALSE; /* flag indicates whether to update the section after command executed */
        zwInfo(zwlog_regen) << Keyword;

        if (!strcmp(Keyword, "CdPlay", 6)) { ... }
        /* rollback history to user-specified feature */
        else if (VxStrSame(Keyword, "CdRollBack") || VxStrSame(Keyword, "CdRelocate")) { ... }
        else if (VxStrSame(Keyword, "CdRollBackAndPlay")) { ... }
        else if (!strcmp(Keyword, "CdHist", 6))
        {
            CdXnWatcher xnWatcher;
            if (VxStrSame(Keyword, "CdHistEditNoRollback")) { ... }
            /* rollback history to user-specified feature and then edit it */
            else if (VxStrSame(Keyword, "CdHistEdit")) { ... }
            /* reorder history operation */
            else if (VxStrSame(Keyword, "CdHistReorder")) { ... }
            /* close all expanded history operations */
            else if (VxStrSame(Keyword, "CdHistCloseAll")) { ... }
            /* update all history operations to current version */
            else if (VxStrSame(Keyword, "CdHistVersionUp")) { ... }
            /* set "auto reduce" flag for all history operations that support it */
            else if (VxStrSame(Keyword, "CdHistAutoReduce")) { ... }
            else if (!strcmp(Keyword, "CdHistEvalAt", 12)) { ... }
            /* clear entity attributes */
            else if (VxStrSame(Keyword, "CdHistClearAt")) { ... }
            else { ... }
        }
    }
}
```

播放

回滚

重定义

移动特征

特征流程——特征执行入口

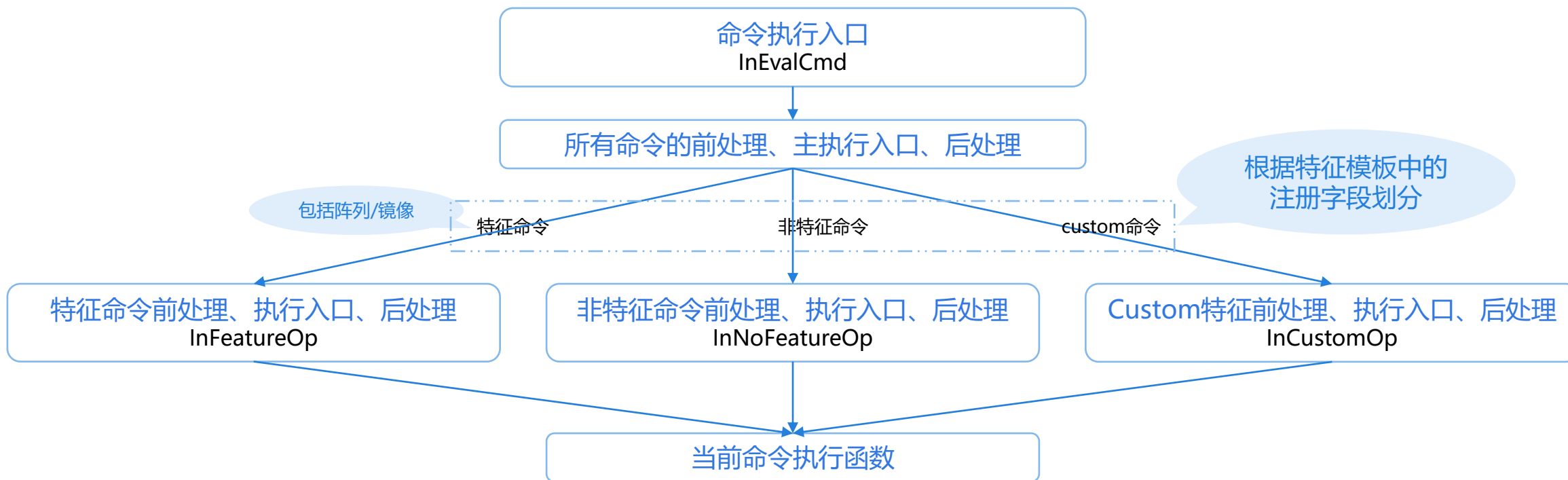
特征执行流程

特征执行的命令流程

InErrEval (3000之前)：特征命令和一些非特征命令的统一入口

InEvalCmd (3000及之后)：InErrEval进行了流程规范的版本

有一个VgFtrTypMgr可以查询当前特征的状态（创建、重生成、重定义、echo etc）



模板命令、特征参数——定义一个特征

模板命令

模板命令 [3]

模板命令：拥有规定的UI、输入限制、执行逻辑的一个功能明确的特征命令

tcmd：template command的缩写，一个文件格式，用于定义特征的参数（类型、限制etc）、特征类型、特征执行命令等的文件

tcmd文件结构（节选自FtAllExt.tcmd）

```
<?xml version="1.0" encoding="UTF-8"?>
<templates xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../schemas/Template.xsd">
  <template name="FtAllExt">
    <property name="icon_id">11</property>
    <property name="function">FtAllExt</property>
    <parameters>
      <parameter luid="14" description="Combine method" type="option">
        <property name="options">@sym_int=0,|enable=FtBoolHasShape,|auto_log,|reactivate,|echo_at_last</property>
        <property name="callback">FtAllExtCb</property>
      </parameter>
      <!-- ..... -->
      <parameter luid="31" description="Extrude type" type="option">
        <property name="options">@sym_int=1,|auto_log,</property>
        <property name="callback">FtAllExtSkip</property>
      </parameter>
    </parameters>
    <!-- XXX is stored in field 9 -->
  </template>
</templates>
```

tcmd由（1）template、（2）template的property、（3）parameter以及（4）parameter的property，共4个部分组成。

ZW3D右下角输入\$report再点击命令，可以看到当前命令的模板名，就能找到对应的tcmd文件，进而找到命令主函数
一般模板名和函数名相同

模板命令、特征参数——定义一个特征

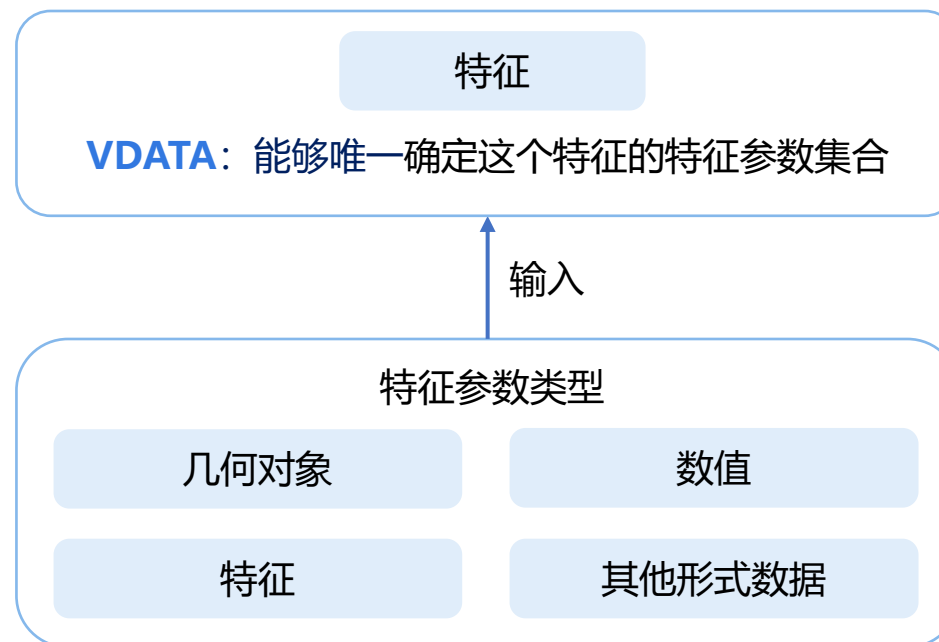
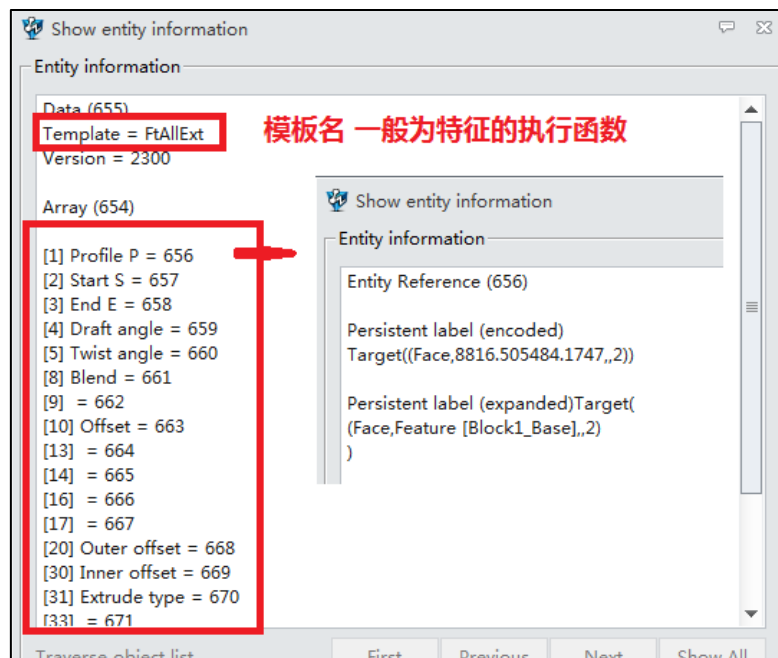
VDATA

VDATA

VDATA

含义1：模板命令参数的实际数据称为VDATA。数据类型为VIDX_LIST（数组），存储了每个参数的索引

含义2：抽象意义的特征参数集合



Brep数据管理——特征执行时模型数据处理

Tt和Brep数据模式

(3000版本之前) Tt和Brep数据模式^[4]

Tt: 拓扑表TopTable缩写, 命令**执行建模**时的存储模型数据的一个表

Tt对象: 拓扑表中存储的面、边、体、环等模型数据对象, 数据名格式为**VsXx**

Brep: **持久化**地保持在文件中的模型数据, 和Tt的结构一致

Brep对象: 持久化存储的面、边、体、环等几何、拓扑数据对象, 数据名格式为**VsCdXx**

```
typedef struct VsApplTopTable
{
    /* Topology information. */
    VsListObj *tbl_shape; /* List of VsShape structures. */
    VsListObj *tbl_shell; /* List of VsShell structures. */
    VsListObj *tbl_comp; /* List of VsCompartment structures. */
    VsListObj *tbl_loop; /* List of VsLoop structures. */
    VsListObj *tbl_edge; /* List of VsEdge structures. */
    VsListObj *tbl_vrtx; /* List of VsVertex structures. */
    /* Geometry information. */
    VsListObj *list_geom; /* List of VsLocGeomDat structures,
    * local NURB geometric entities */
    /* Database information. */
    VsListObj *list_idx; /* List of VsDbIdxObj structures,
    * database indices for geometric
    * and topological entities */
    void *tbl_xref; /* Cross reference table VsFcXRefTable */
} VsApplTopTable;
```

```
typedef struct
{
    int idx_shapes; /* head of list of shape objects */
    int idx_shells; /* head of list of shell objects */
    int idx_loops; /* head of list of loop objects */
    int idx_edges; /* head of list of edge objects */
    int idx_vertices; /* head of list of vertex objects */
    int idx_geom; /* head of list of geometry objects */
    int last_ftr; /* last active feature index */
    int last_shell; /* last shell position code */
    int id_shell; /* counter for generating shell pick id's */
    int id_face; /* counter for generating face pick id's */
    int id_edge; /* counter for generating edge pick id's */
    int idx_par; /* index of parent object */
    int i1; /* spare integer */
    int i2; /* spare integer */
} VsCdBrep;
```

- 1、模型越大数据越多, 耗时越长
- 2、Tt相当于草稿纸, 可以随时废弃重来

准备拓扑表 (Brep - Tt)
CdTargUpdTt

获取输入

建模

更新Brep (Tt - Brep)
CdTargUpdBrep

特征执行流程

Brep数据管理——特征执行时模型数据处理

od对象和Zw对象

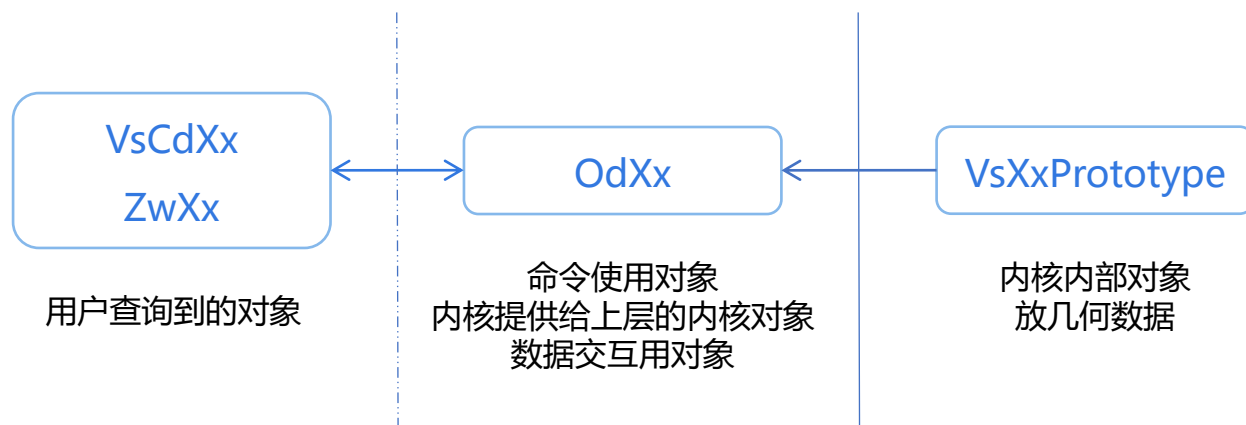
(3000版本及之后) Od对象和Zw/VsCd对象

Tt <=> Brep数据模式问题

- 1、单brep结构，拓扑表结构颗粒度过大，每次只能全量加载和更新，无法支持长历史大模型的图纸，越往后单个特征耗时越长
- 2、其他：模块耦合、无法选择性回滚实体状态etc

Partition模式：以体（body）为单位，一个partition一个体，一个模型多个partition的形式，划分模型数据

数据隔离：由原先的VsCd对象既存几何拓扑数据，也存应用数据，变成VsXx对象存几何数据由内核管理，其他数据上层管理



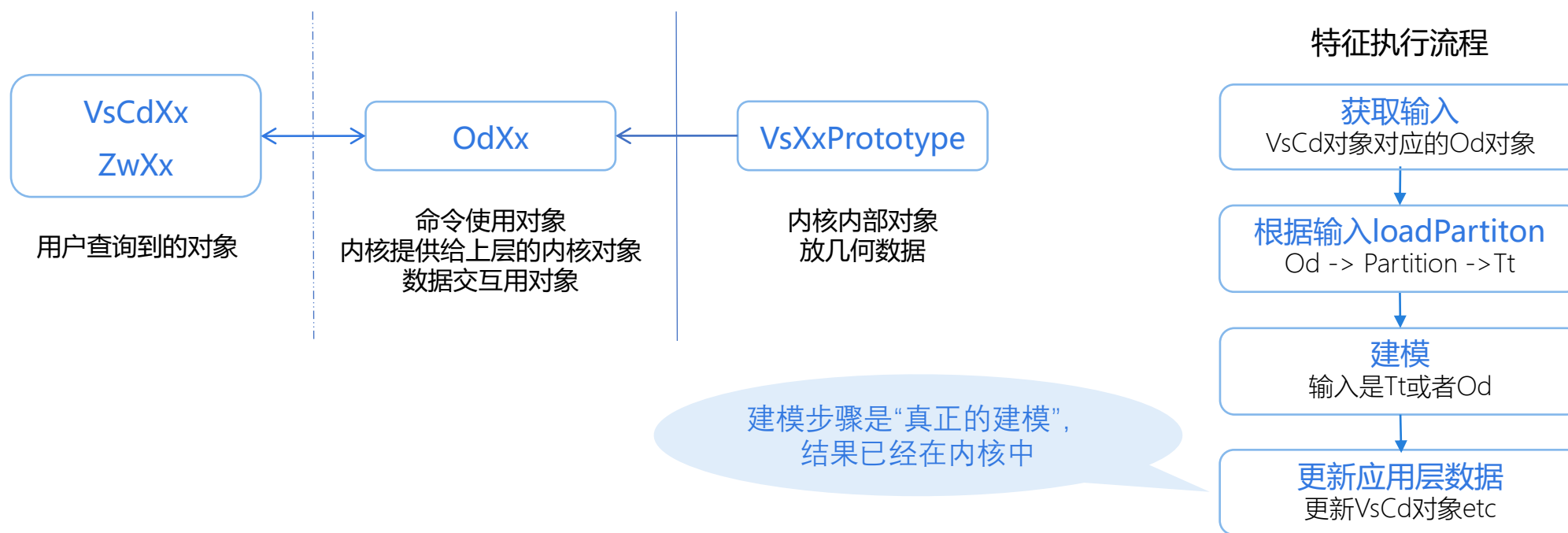
Brep数据管理——特征执行时模型数据处理

od对象和Zw对象

(3000版本及之后) Od对象和Zw/VsCd对象

Partition模式：以体 (body) 为单位，一个partition一个体，一个模型多个partition的形式，划分模型数据

数据隔离：由原先的VsCd对象既存几何拓扑数据，也存应用数据，变成VsXx对象存几何数据由内核管理，其他数据上层管理



永久命名——（特征结果）实体对象的唯一标识

永久命名意义和作用

永久命名label^[5]

场景：

- 1、对一个Block特征的一条边做圆角，重生成整个历史的时候，怎么找到这条边，应该记录什么？
- 2、对中间某个特征A进行改参，后续有特征B依赖了这个特征A改参前产生的一个面，特征B该记什么才能在改参后再次找到这个面？

永久命名——（特征结果）实体对象的唯一标识

永久命名意义和作用

永久命名label

场景：

- 1、（不改参重生成）有两个Block特征A和B，对BlockA的一条边做圆角，重生成整个历史的时候，怎么找到这条边，应该记录什么？
- 2、（改参重生成）对拉伸特征A进行改参，后续有特征B依赖了这个拉伸特征A改参前产生的一个面，特征B该记什么才能在改参后再次找到这个面？

解答：

- 1、把BlockA这个信息记录下来，同时在BlockA中，由两个面确定这个唯一的边，把边的邻面信息也记下来
- 2、赋予拉伸体的面一些语义，比如拉伸的底面、顶面，把这个工程语义也记录下来

建模顺序也是一种信息



永久命名

永久命名用于唯一定义实体的，每一个实体拥有一个名字（label）

一般而言，这个名字需要记录一些该实体被创建时的一些参数信息，或者拓扑信息

属性^[6]

属性

表示附着对象的所具有的某种性质，如外观属性（颜色、显隐性、透明度、图层etc）、工程意义属性、用户自定义属性

属性附着对象

绘图区用户可以拾取到的对象

属性特点

非时序性的，附着对象存在即生效

属性机制

- 1、属性数据管理：多种属性，多种附着对象导致数据管理比较复杂
- 2、属性传播：当附着对象发现变化时，不同的属性有不同的传播要求（比如面A分裂成C和B，要求原来A上属性同时传播到C和B上）
- 2、属性刷新^[6]：模型重生成时有特殊的刷新机制让属性找到附着对象

其他机制

属性、静态

静态机制

静态实体定义

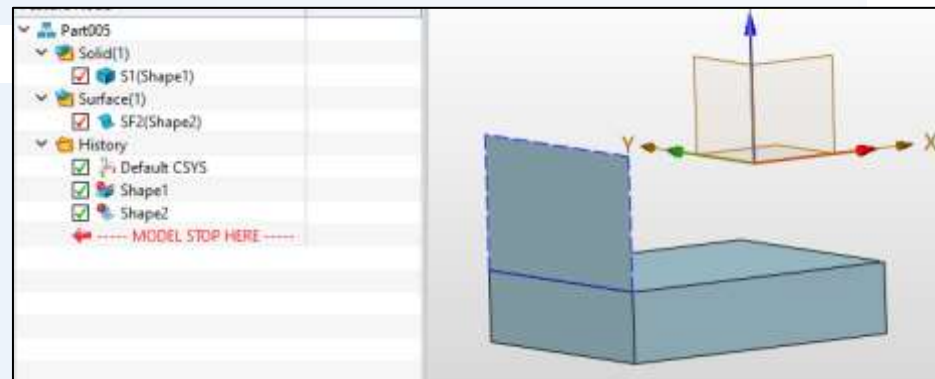
在一些场景下，出于保密等考虑，经常需要删除一个模型的驱动参数而仅保留模型结果，让这个模型失去参数化的能力，因此需要拥有处理“非参数驱动几何体”的能力，在程序中体现为静态实体。（此处仅指体和片体，即shell）

创建静态实体

Defeature去参：选择部分特征去参，将这部分特征的模型结果变成静态实体

Encapsulate封装：将当前模型全部去参，变成静态实体

导入几何：只导入模型结果，产生静态实体



静态实体的label
也需要特殊处理

静态特征和静态实体

ZW3D历史树上一个静态特征对应一个静态实体

其他机制

属性、静态

静态机制

静态特征

无参数驱动，无法根据参数重生成、无法改参，只有结果
是对历史连续性的破坏

静态机制（3000之前）^{[8][9]}

为了支持静态特征的模型被后续特征改参，且回滚后能够获得最初的模型，会将最初的静态模型数据进行备份，用于让静态特征能够在正确地“重生成”
这种备份和特殊处理将会穿插在所有特征相关的任何地方

静态机制（3000及之后）

在partition模式下，静态特征被处理成：静态实体是当前特征生成的，对应的partition从空的pmark0状态到有静态实体的pmark1。
静态特征播放的时候是从静态实体所属的partition的pmark0跳转到pmark1，回滚则是pmark1跳回pmark0。行为和其他特征行为一致，不再需要特殊处理，特征行为的逻辑统一



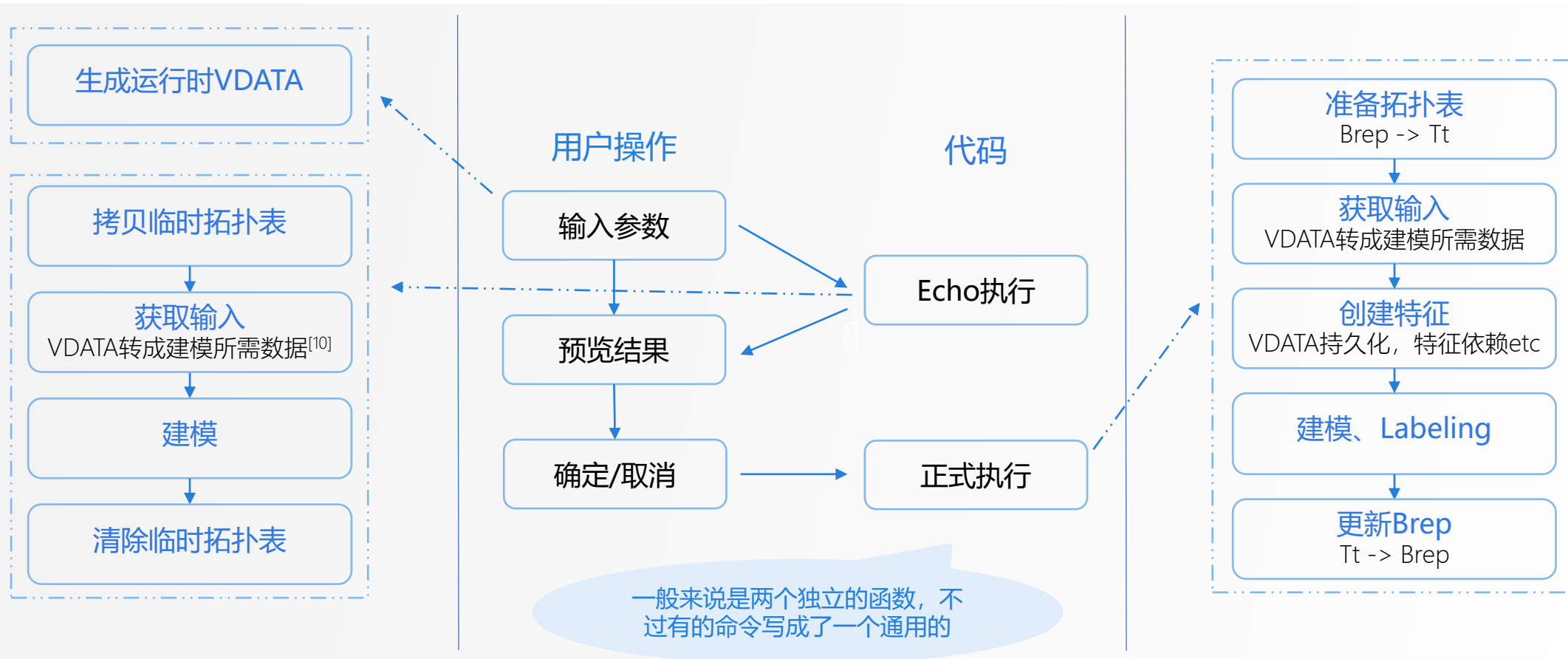
特征操作（不同时期）实现介绍

- 正向建模
- 回滚
- 播放
- 重定义
- 重生成

正向建模

echo执行、正式执行

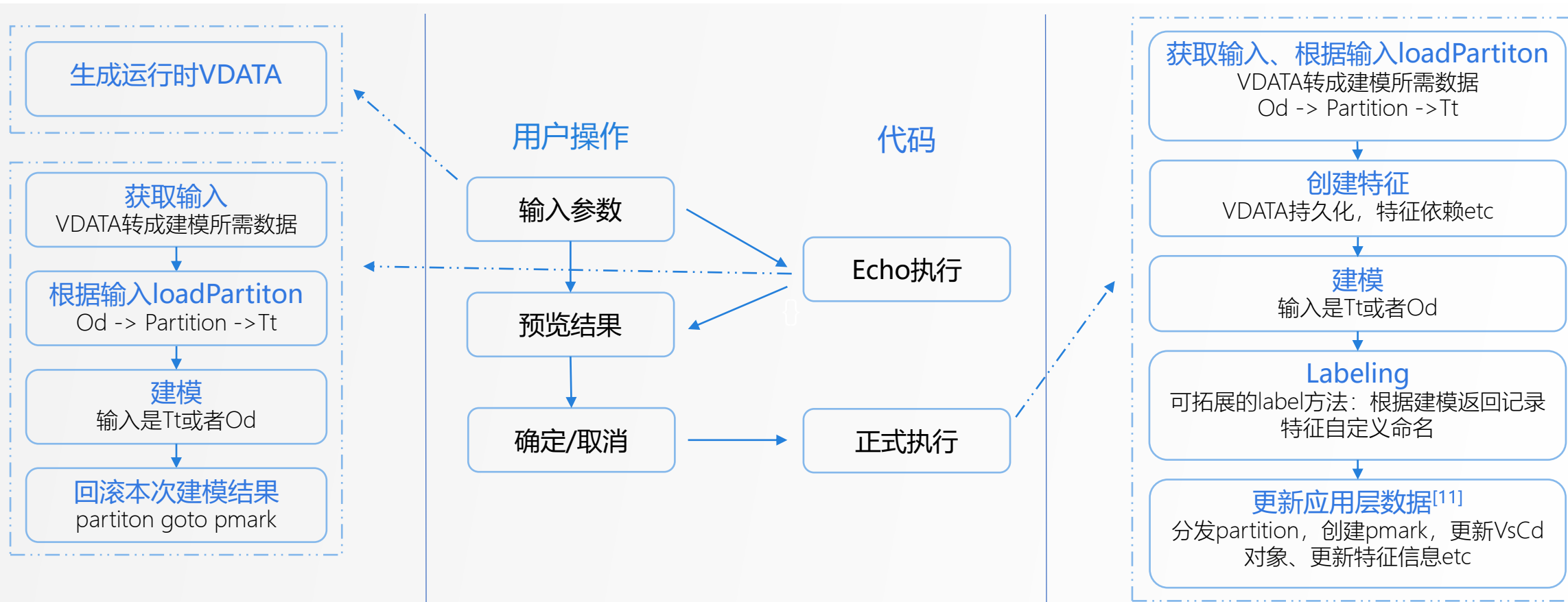
创建特征（3000之前）



正向建模

echo执行、正式执行

创建特征（3000及之后）



回滚特征（3000之前）

含义：回到第n个特征执行前的模型状态

用户触发方式：拖动建模条往上、重定义、抑制/反抑制、删除

方式：快速回滚（HQR）^[12]，若快速回滚失败则进行全量重生成

在没有快速回滚之前是
直接全量重生成

快速回滚

在特征中记录了执行前后Brep和其他历史数据的增删改，回滚时根据这个备份进行状态还原（增量更新）

历史数据的undo-redo
因为此前内核没有事务能力，只能
外部控制回滚

全量重生成

从头开始执行一遍所有的建模条之前的特征，得到新的几何模型

因为只有一份Brep

回滚特征（3000及之后）

含义：回到第n个特征执行前的模型状态

条件基础：内核具有事务能力、以partition为模型数据单位

方式：根据特征中记录的pmark进行回滚

pmark回滚

特征改变一个partition中shell的状态时，会生成一个对应的pmark存在特征中（改变n个partition就生成n个pmark），特征回滚时只要读取保存的pmark，回滚到这个pmark之前的状态即可

播放

建模条拖动

播放特征（3000之前）

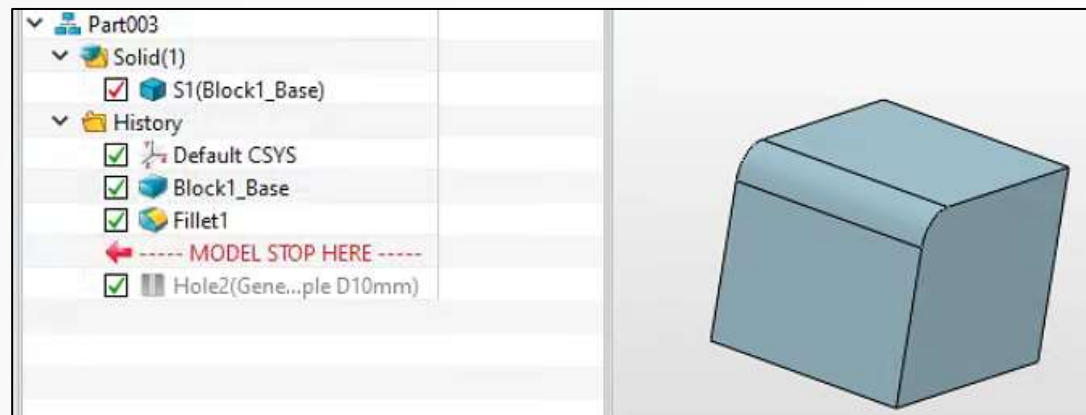
含义：播放历史树上在建模条下面的特征

用户触发方式：同回滚

方式：特征重生成

特征重生成

特征记录了用户的输入，重生成是根据用户输入重新求解模型



播放选项：播放下一个、播放所有.....

播放

建模条拖动

播放特征（3000及之后）

含义：播放历史树上在建模条下面的特征

用户触发方式：同回滚

方式：特征重生成regen、rollforward

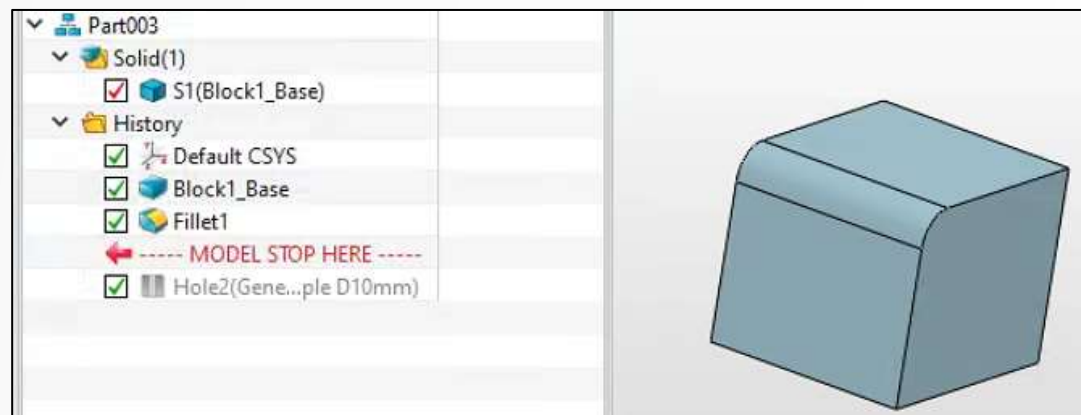
特征重生成

特征记录了用户的输入，重生成是根据用户输入重新求解模型

rollforward

这里以后会变成一个新的“计算管线”
机制，后续有课程介绍

当特征依赖判断到前面发生改变的特征，和当前播放的特征没有任何关系的时候，我们可以直接跳转pmark到这个特征上一次执行后的模型状态



播放选项：播放下一个、播放所有.....

特征重定义

改参

重定义

含义：通过修改特征参数实现修改模型对应部分几何

用户操作：双击特征/右键选择重定义命令

方式：**回滚**到前一个特征 - 重新获取输入 - 重新**执行**这个特征 - **重生成**下面的特征

删除特征是类似的方式：
回滚到前一个特征 - 移除选中特征 - 播放剩下的特征

特征重生成

重新执行特征

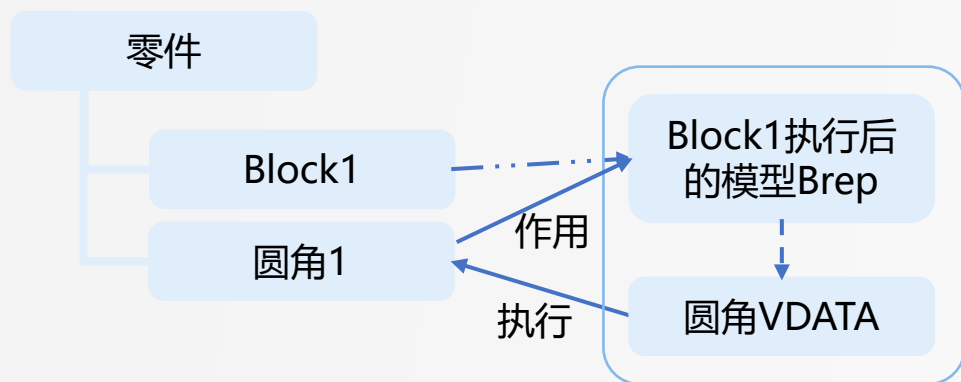
重生成^{[13][14]}

含义：按照VDATA记录的用户输入重新执行这个特征

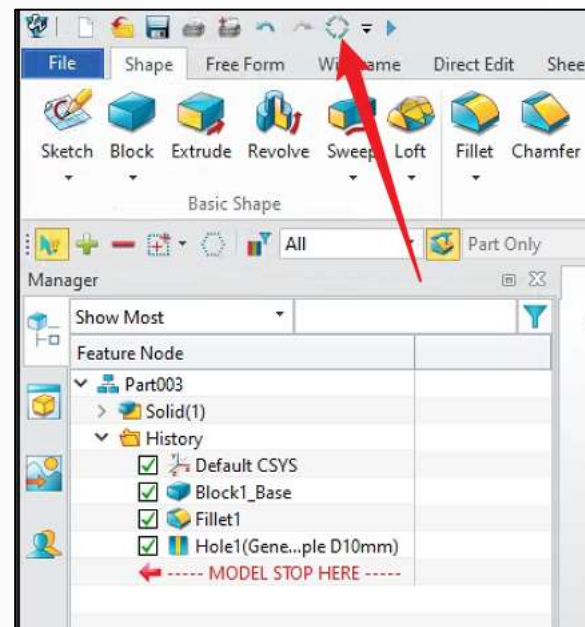
会触发重生成的用户操作：删除特征、抑制/反抑制特征、重定义特征，主动拖动建模条播放特征，主动全部重生成，回滚时快速回滚失败，修改标注，删除静态对象.....

特征需要重生成的原因：（输入的改变）特征的VDATA改变

（输入/作用对象的改变）前面特征对应的模型Brep状态改变



严格来讲应该是，
特征作用的Brep
对象改变



重生成模型按钮

参考文献



注1：文档是深入介绍，请按需阅读，绝大部分后续还有相应课程再介绍

注2：文档链接为相关概念的个人推荐阅读文档，部分细节概念本次没有介绍则没有附上链接。参数化有较多的文档，可按需搜关键词查阅

- [1] 《特征依赖概述》 <https://zwiki.zwcax.com/pages/viewpage.action?pagelId=90046770>
- [2] 《Custom特征概述》 <https://zwiki.zwcax.com/pages/viewpage.action?pagelId=110338709>
- [3] 《模板命令开发流程》 <https://zwiki.zwcax.com/pages/viewpage.action?pagelId=129499450>
- [4] 《拓扑表结构》 <https://zwiki.zwcax.com/pages/viewpage.action?pagelId=103842206>
- [5] 《Labeling》 <https://zwiki.zwcax.com/display/Platformarchitecture/Labeling>
- [6] 《实体属性概述》 <https://zwiki.zwcax.com/pages/viewpage.action?pagelId=110338653>
- [7] 《属性刷新机制》 <https://zwiki.zwcax.com/pages/viewpage.action?pagelId=110338731>
- [8] 《静态Brep的定义与表达》 <https://zwiki.zwcax.com/pages/viewpage.action?pagelId=90046814>
- [9] 《静态对象备份机制》 <https://zwiki.zwcax.com/pages/viewpage.action?pagelId=103842204>
- [10] 《VDATA解析与持久化》 <https://zwiki.zwcax.com/pages/viewpage.action?pagelId=90046776>
- [11] 《Partition&Brep management》 <https://zwiki.zwcax.com/pages/viewpage.action?pagelId=120593009>
- [12] 《快速回滚实现逻辑》 <https://zwiki.zwcax.com/pages/viewpage.action?pagelId=90046760>
- [13] 《重生成流程》 <https://zwiki.zwcax.com/pages/viewpage.action?pagelId=103842139>
- [14] 《草图重生成》 <https://zwiki.zwcax.com/pages/viewpage.action?pagelId=103842177>

感谢聆听

Q&A