

ZW³SOFT

特征VDATA框架与参数规则

主讲人：任赞佐

C O N T E N T S

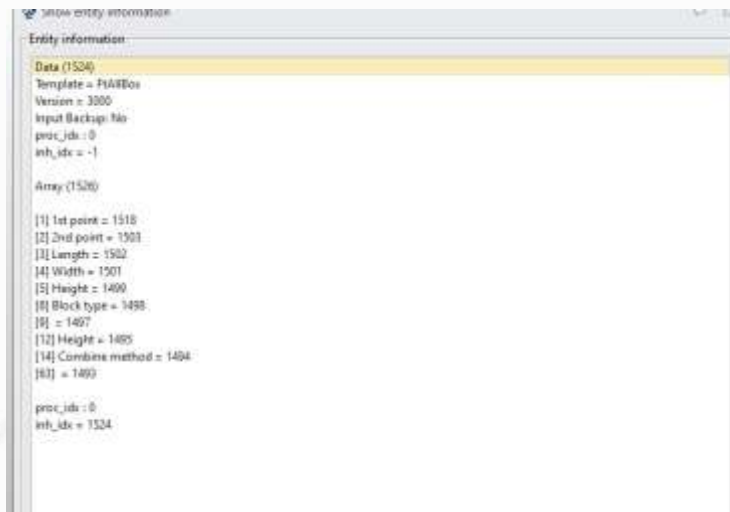
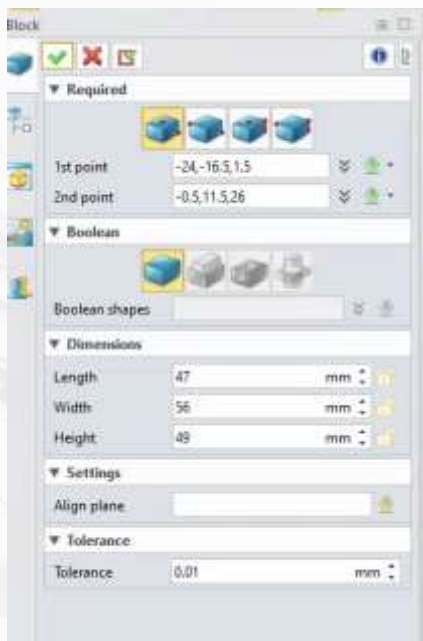


特征参数框架



VDATA使用规范

什么是参数？

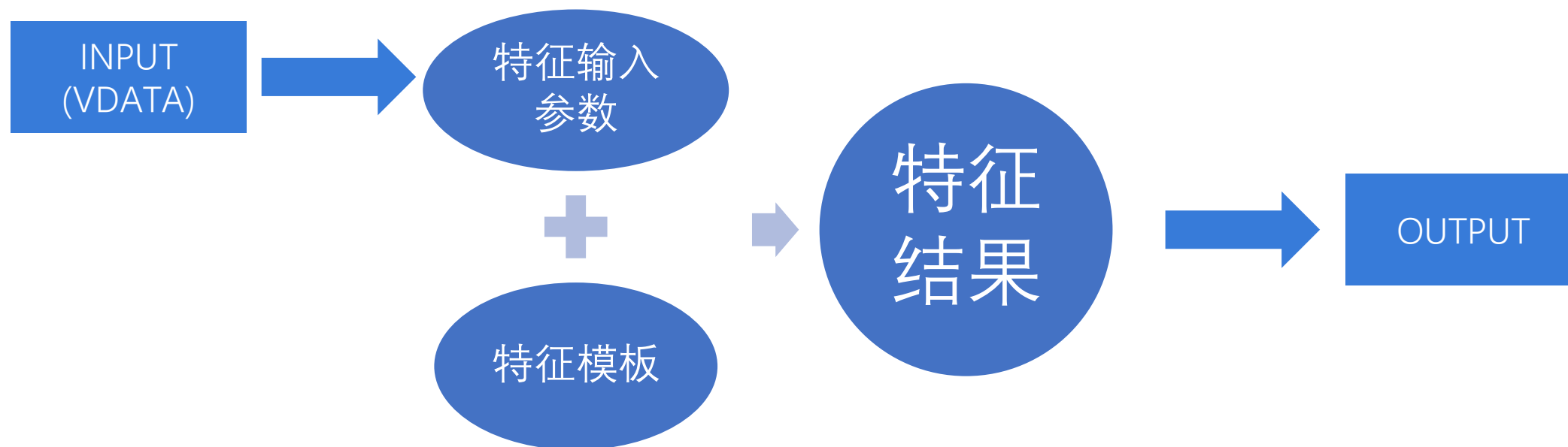


参数就是特征VDATA

参数规则就是特征输入的设计、使用规范

- 1、关于特征参数如何编码，一些最基本的参数准则
- 2、特征的vdata如何设计，对于一些特别的需求如何能够在符合参数化的基础上设计
- 3、特征vdata如何使用，有哪些准则，在编码上有什么要注意的

特征参数模型



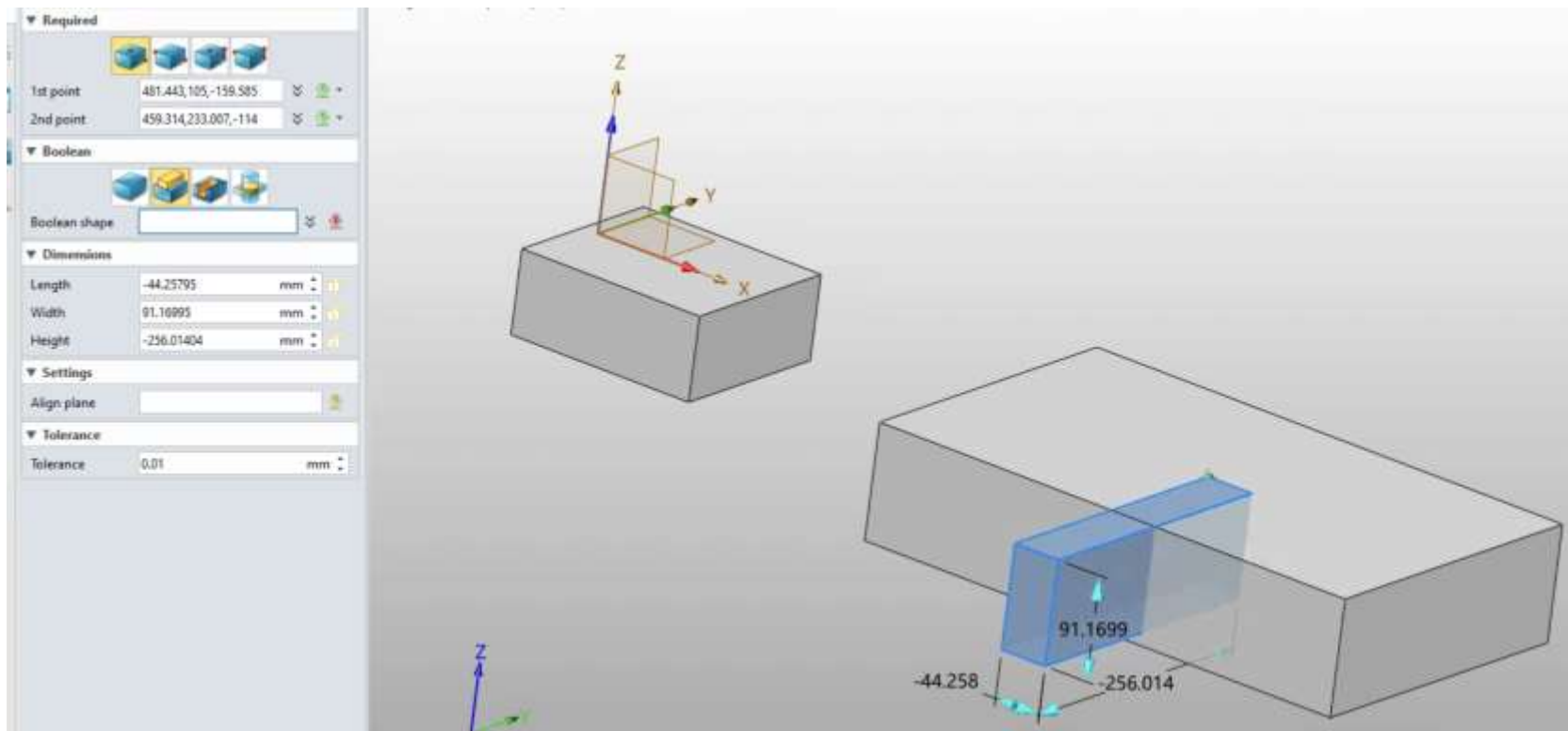
输入和特征使用应该一一对应

1、用了不记

2、记了不用

特征应该保证内部的自洽性、一致性是一个独立自洽的单元

补登记



为什么需要补登记？

它对应的上面哪个原则？

如何决定一个特征的输入？

1、分析需求，实事求是

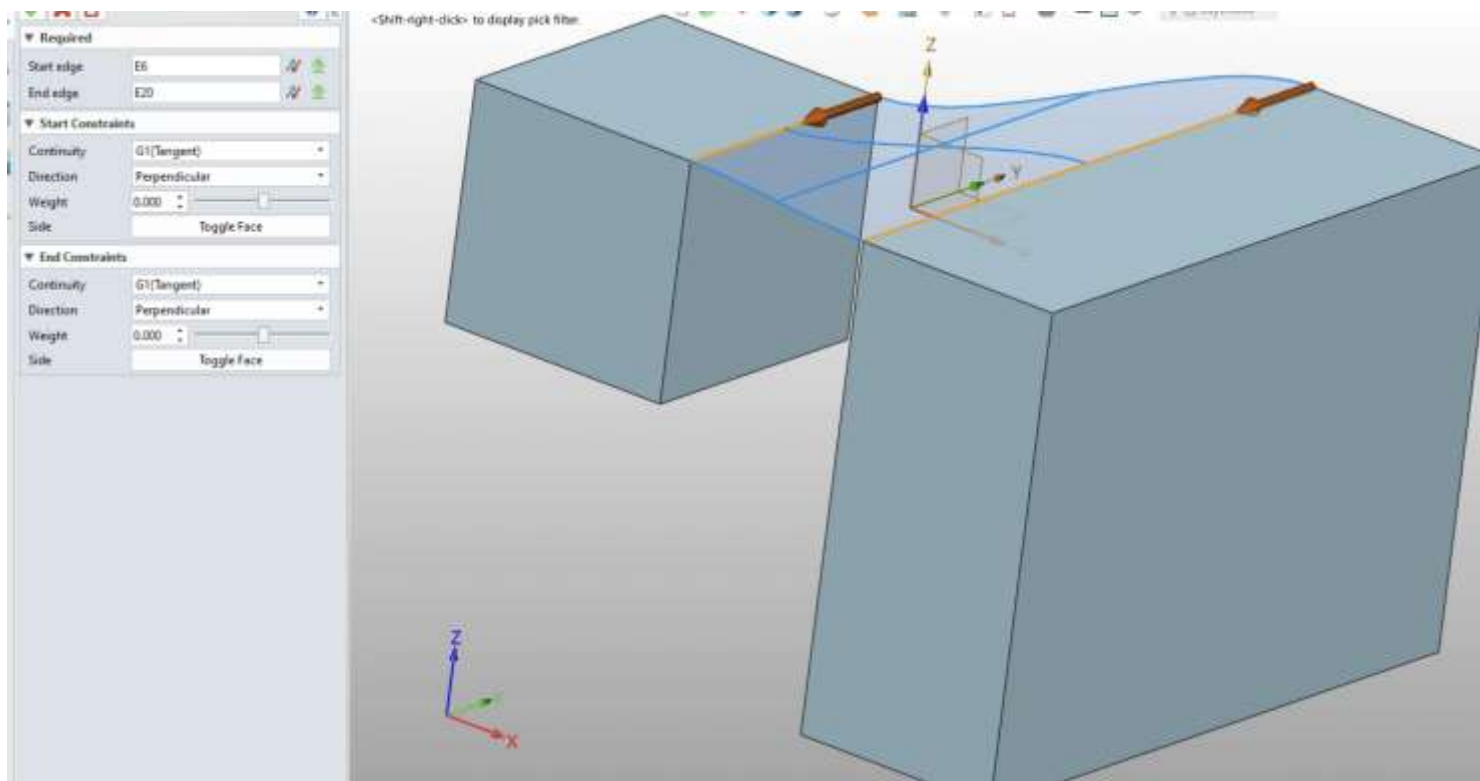
2、了解一些基本组件，复用基本组件

3、特殊需求通过模版实现

1、参考产品

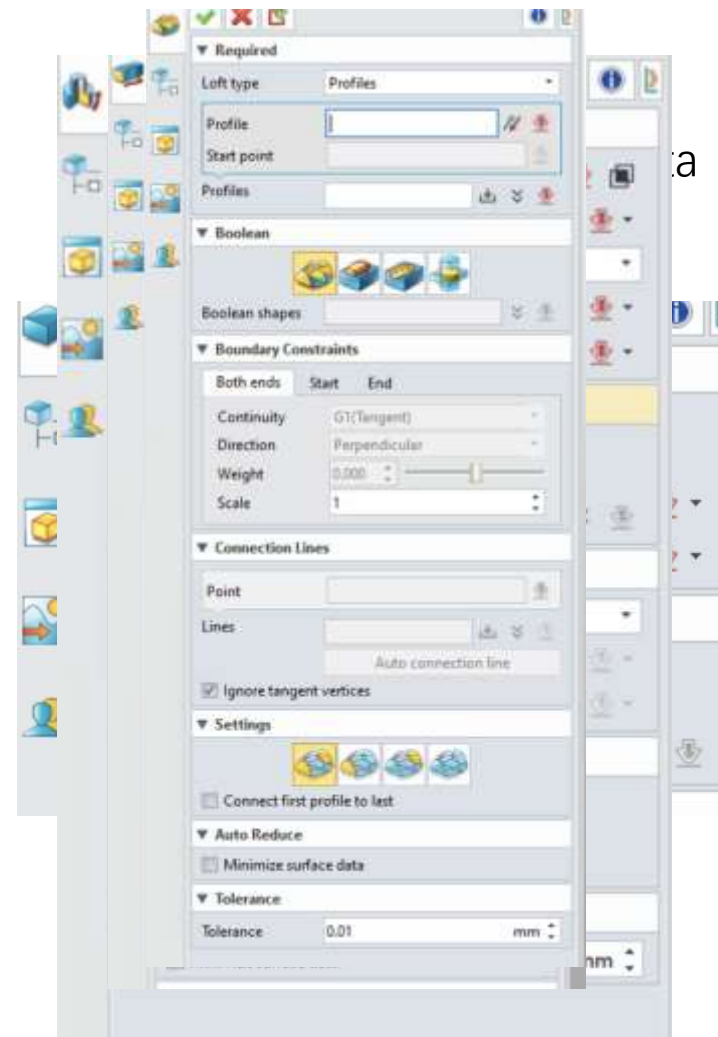
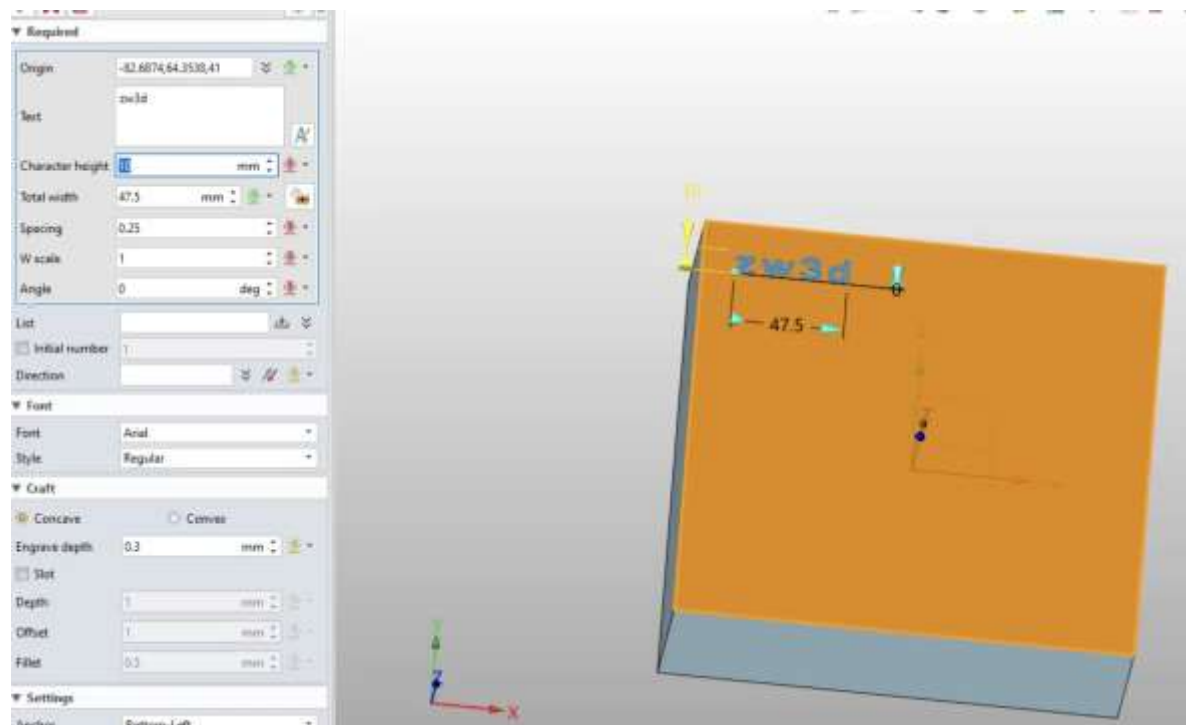


特征参数模型



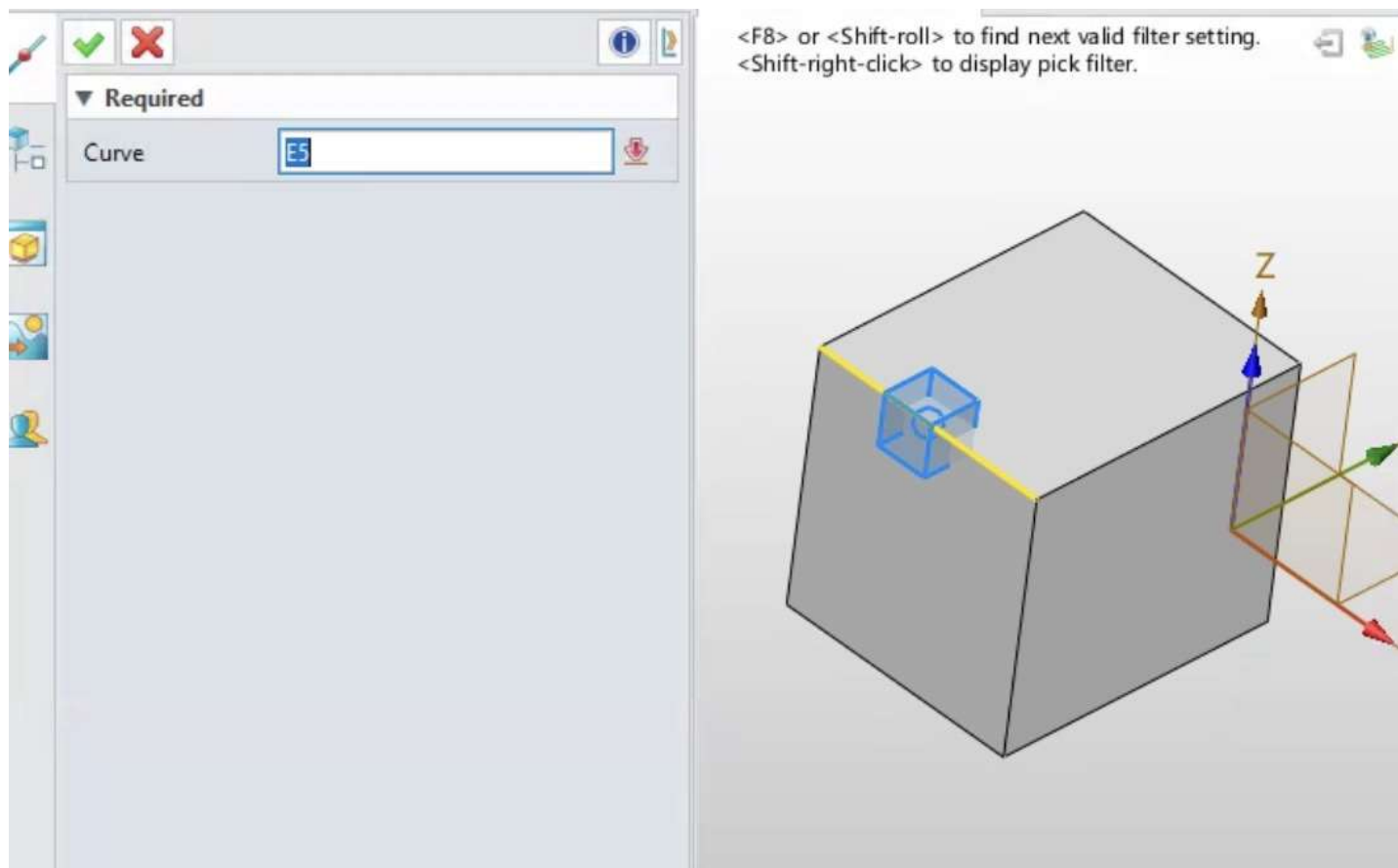
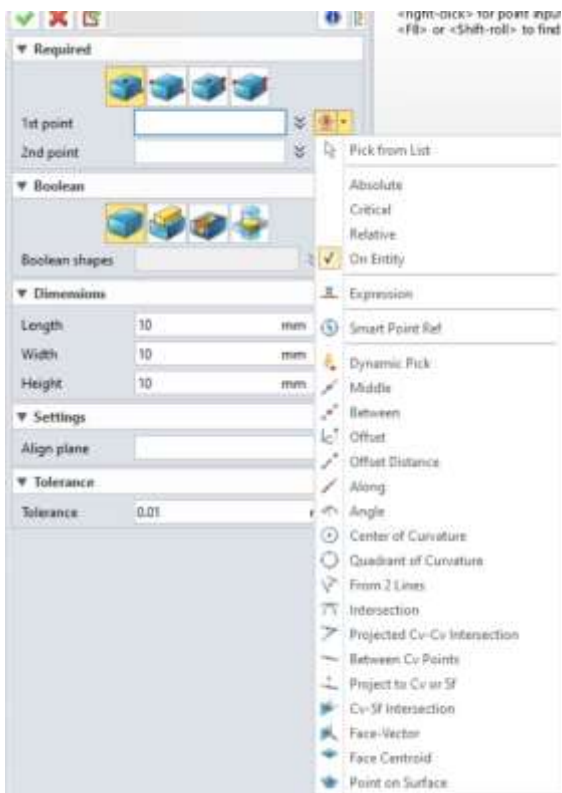
如何设计这个特征的vdata

特征参数模型

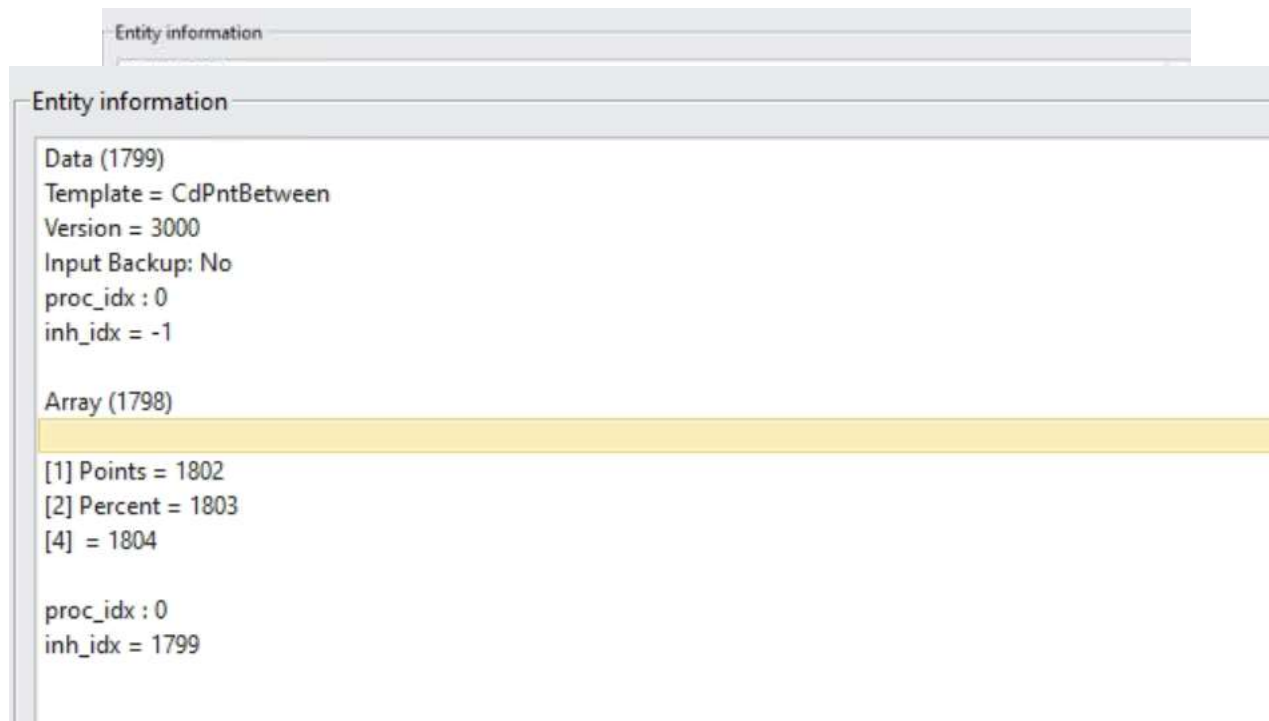
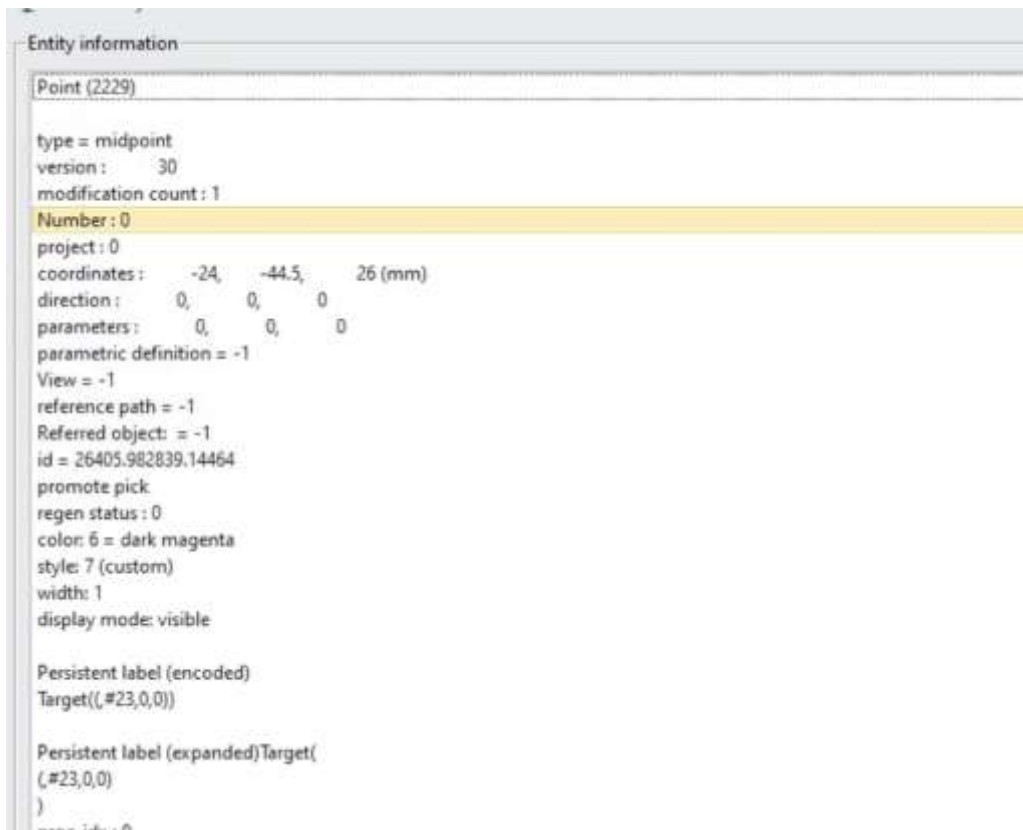


特征参数规则

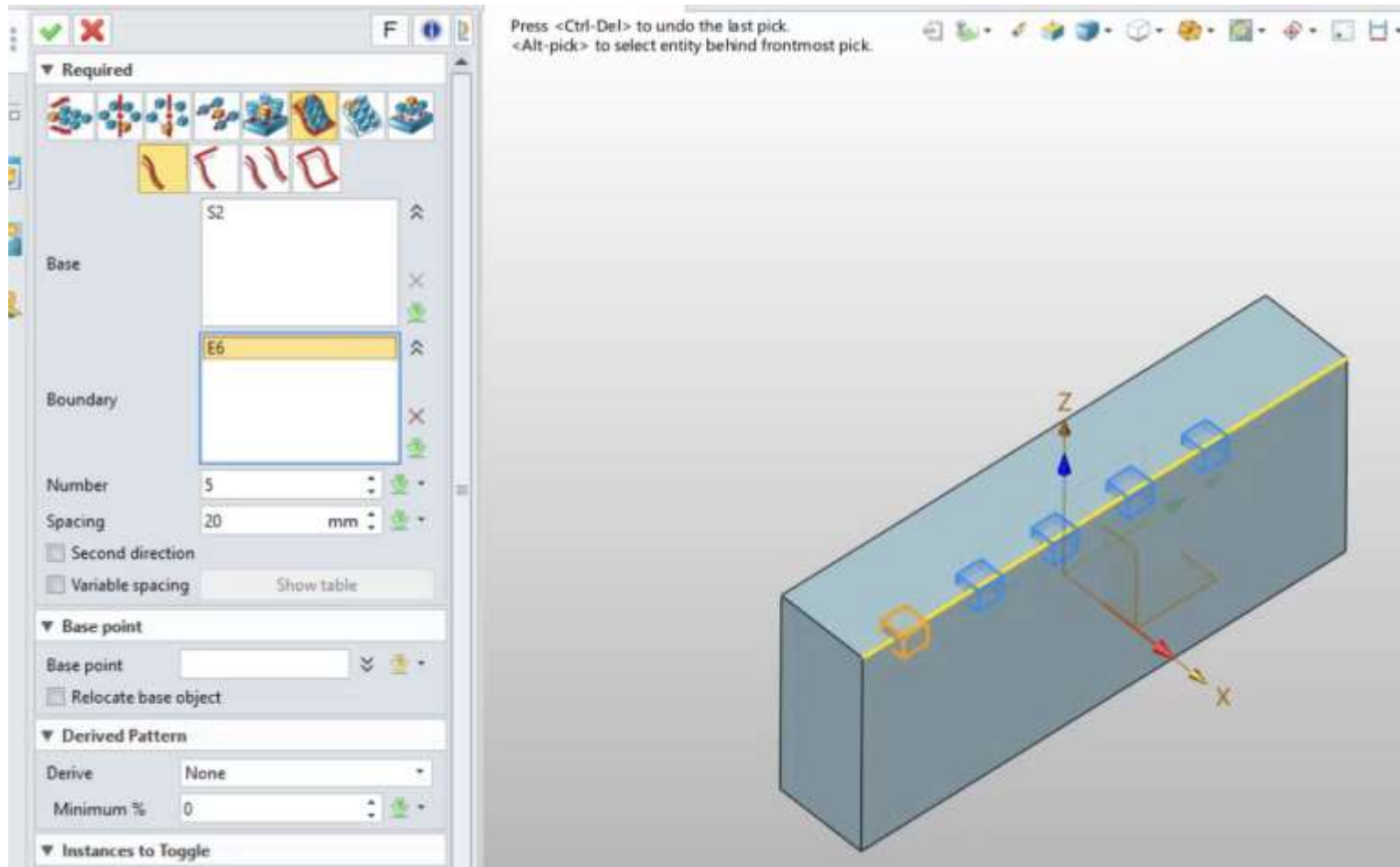
特殊的参数模版



特征参数规则



特征参数模型



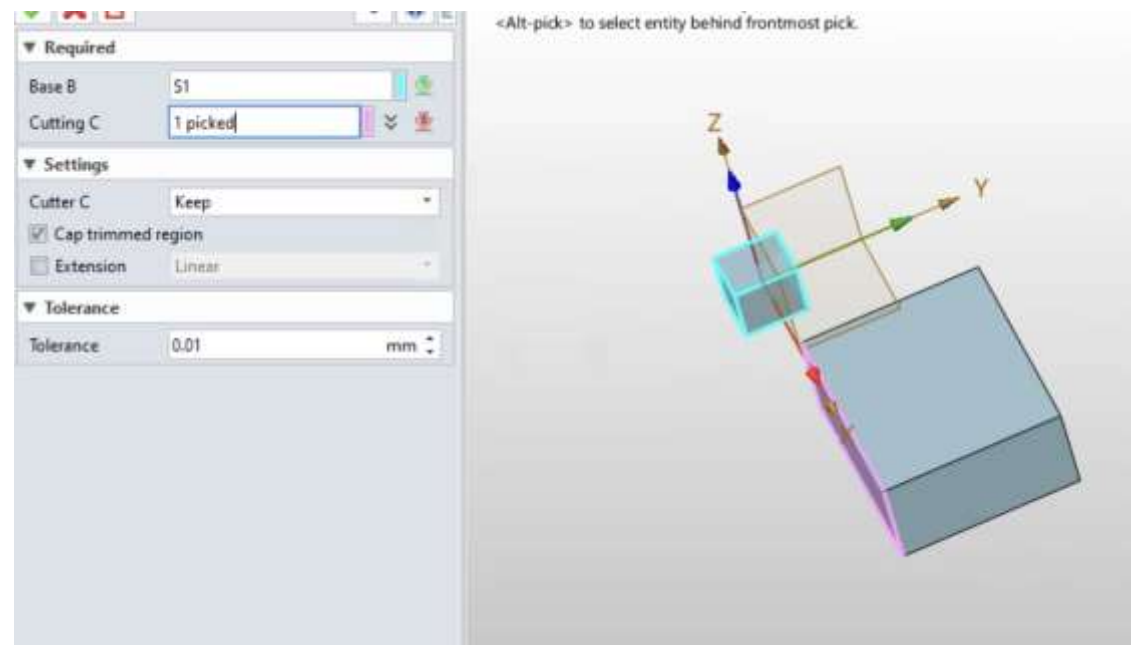
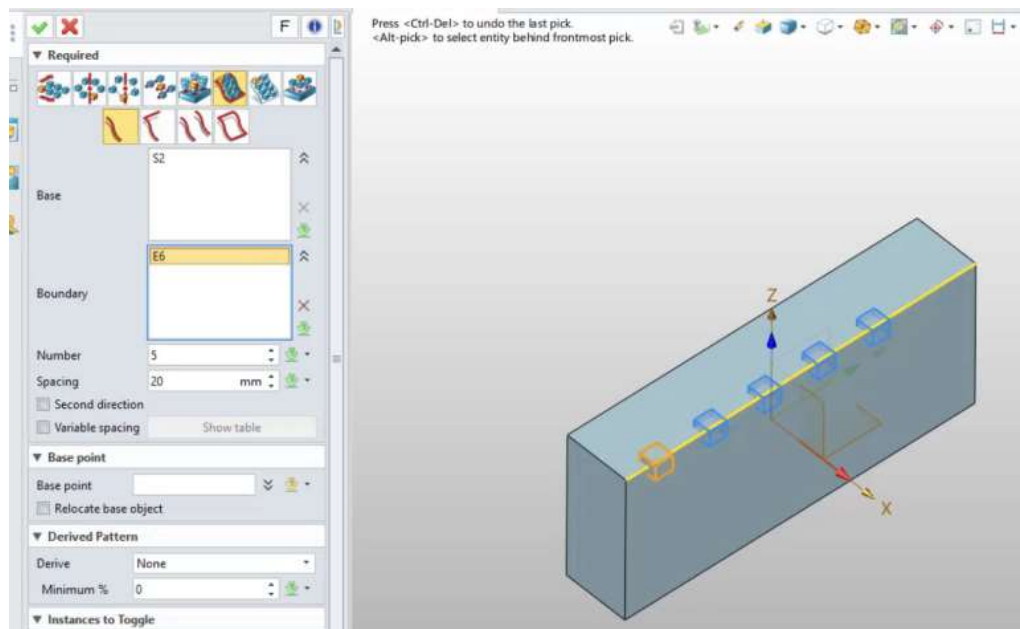
如何设计这个特征的vdata

输入边->特征建模（将拿到的边转换为curve）

参数模版输入为edge->输出curve 作为vdata->特征建模

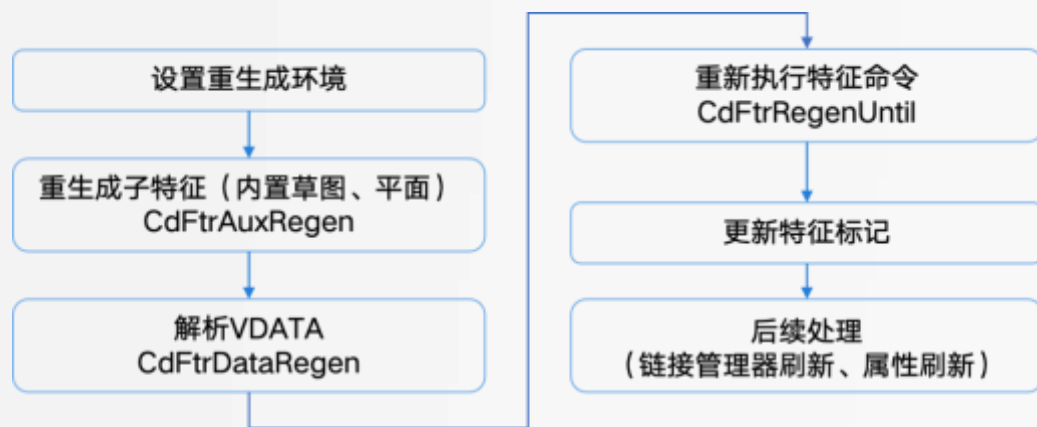
有什么区别？

特征参数模型



可以考虑新增加vdata类型
参考一个几何的情况不应该直接使用实体对象作为输入

特征参数规则



CdFtrRegen流程

```
ZW3D_CAD.dll!CdPntBetween(int idx_in, int * idx_out) Line 39
ZW3D_CAD.dll!InNoFeatureOp(const char * fn_name, int & idx_in, int * idx_out) Line 241
ZW3D_CAD.dll!InDistributeCmd(std::string & fn_name, int fRetryOnErr, int fFormCmdMgr, int & idx_in, int * idx_out) Line 126
ZW3D_CAD.dll!InEvalCmd(std::string fn_name, int fRetryOnErr, int fFormCmdMgr, int & idx_in, int * idx_out) Line 74
ZW3D_CAD.dll!InEval(const std::string & name, int & idx_in, int * idx_out) Line 57
ZW3D_CAD.dll!CdPnt3RegenData(VsObjHandle * oh, VsObjMsg * msg) Line 381
ZW3D_CAD.dll!CdPnt3Regen(VsObjHandle * oh, VsObjMsg * msg) Line 162
ZW3D_Om.dll!OmDocObjMsg(VsObjDoc * doc, VsObjHandle * oh, VsObjMsg * msg) Line 990
ZW3D_Db.dll!FMF::vdata::VsData::FieldRegen() Line 531
ZW3D_Db.dll!FMF::vdata::VsData::Regen() Line 137
ZW3D_Db.dll!FMF::vdata::VsDataMgr::PushDataUtil(int idx_data, VeDataFnType fn_type, VsObjMsg * msg, int bin_idx) Line 955
ZW3D_Db.dll!FMF::vdata::VsDataMgr::PushDataRegen(int idx_data, int bin_idx) Line 861
ZW3D_Db.dll!FMF::vdata::CdDataRegenUtil(int idx_data, int fUpdOrig, int * idx_copy, VsUIdMap * pUIdMap) Line 231
ZW3D_Db.dll!FMF::vdata::CdDataRegenStd(int idx_ftr, int version, int idx_data, int * idx_copy, FMF::vdata::RegenOption & regenOp) Line 147
ZW3D_CAD.dll!CdFtrDataRegen(VsObjHandle * oh, int * idx_in, int * ref_missing, int * MissAutoLog) Line 116
ZW3D_CAD.dll!CdFtrRegen(VsObjHandle * oh, VsObjMsg * msg) Line 357
ZW3D_Om.dll!OmDocObjMsg(VsObjDoc * doc, VsObjHandle * oh, VsObjMsg * msg) Line 990
ZW3D_CAD.dll!CdPlayNext(int stop, int ioption, int clean_data, int * idx_next) Line 282
```

特征参数规则

参数点参数值和特征一样具有vdata以及通过模版求解的结构有同样的要求 - 自洽、独立、正交

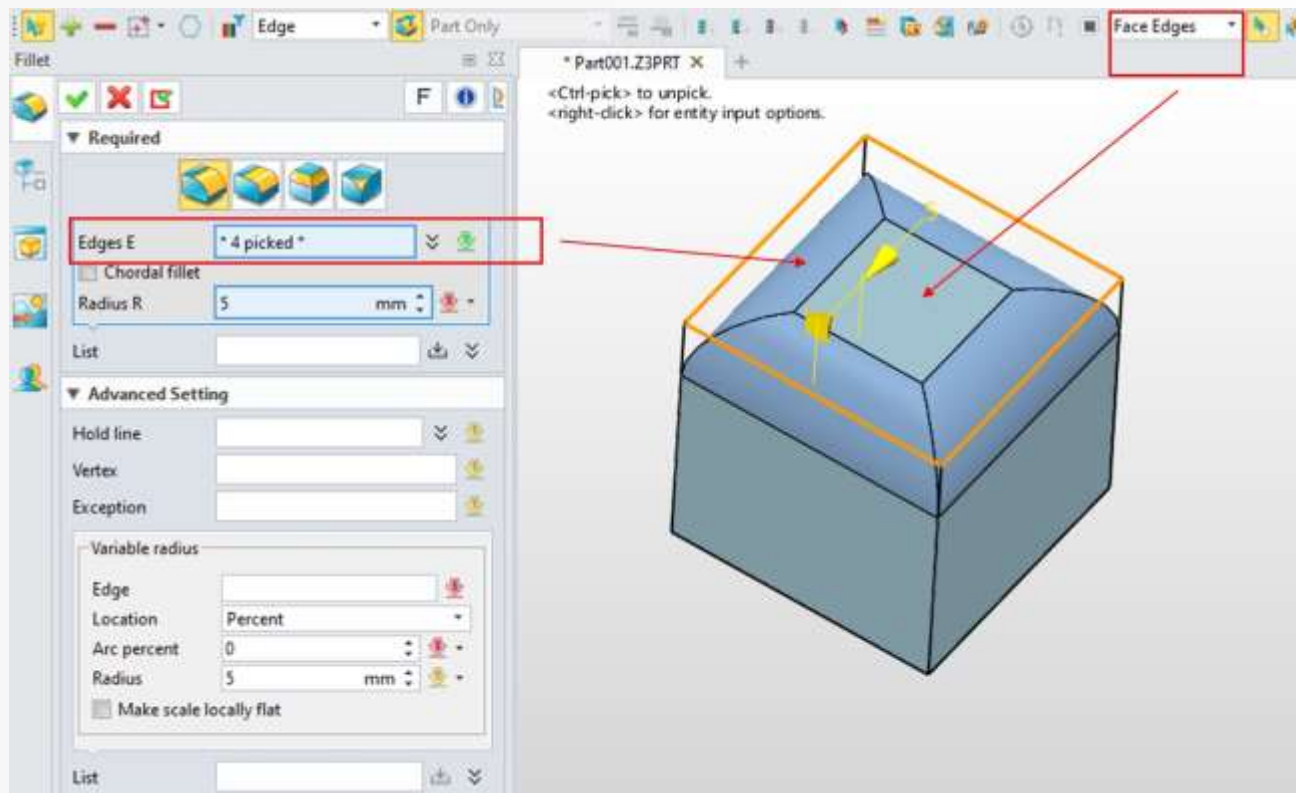
不同点：
output不同

Q：是否所有vdata都有求解的过程，实体对象的求解过程是什么？

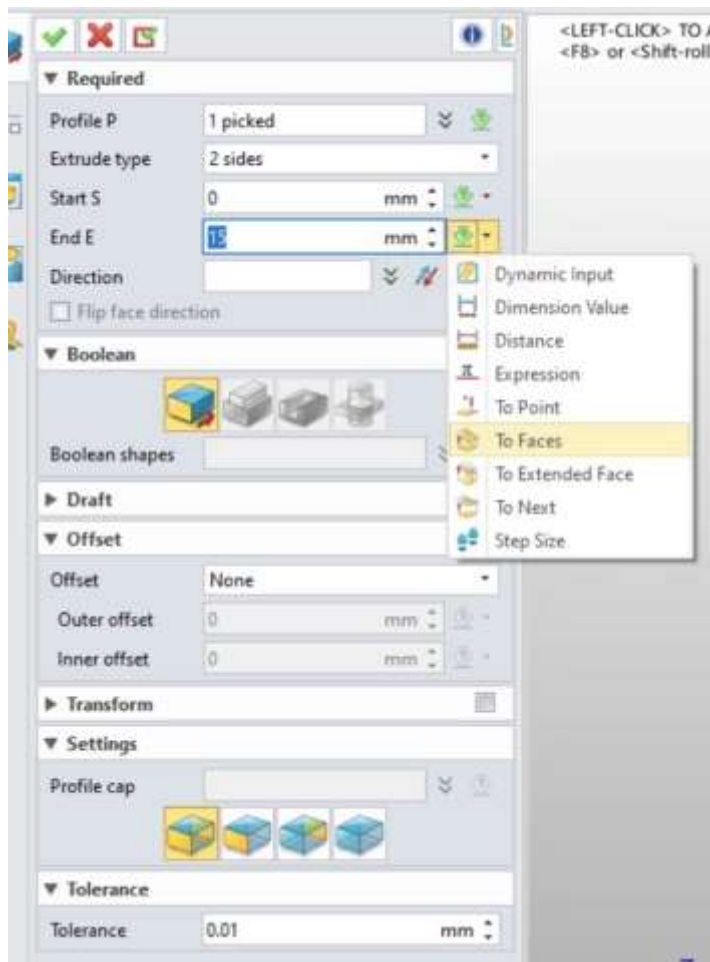
特征参数规则

延伸一下

参数的output种类



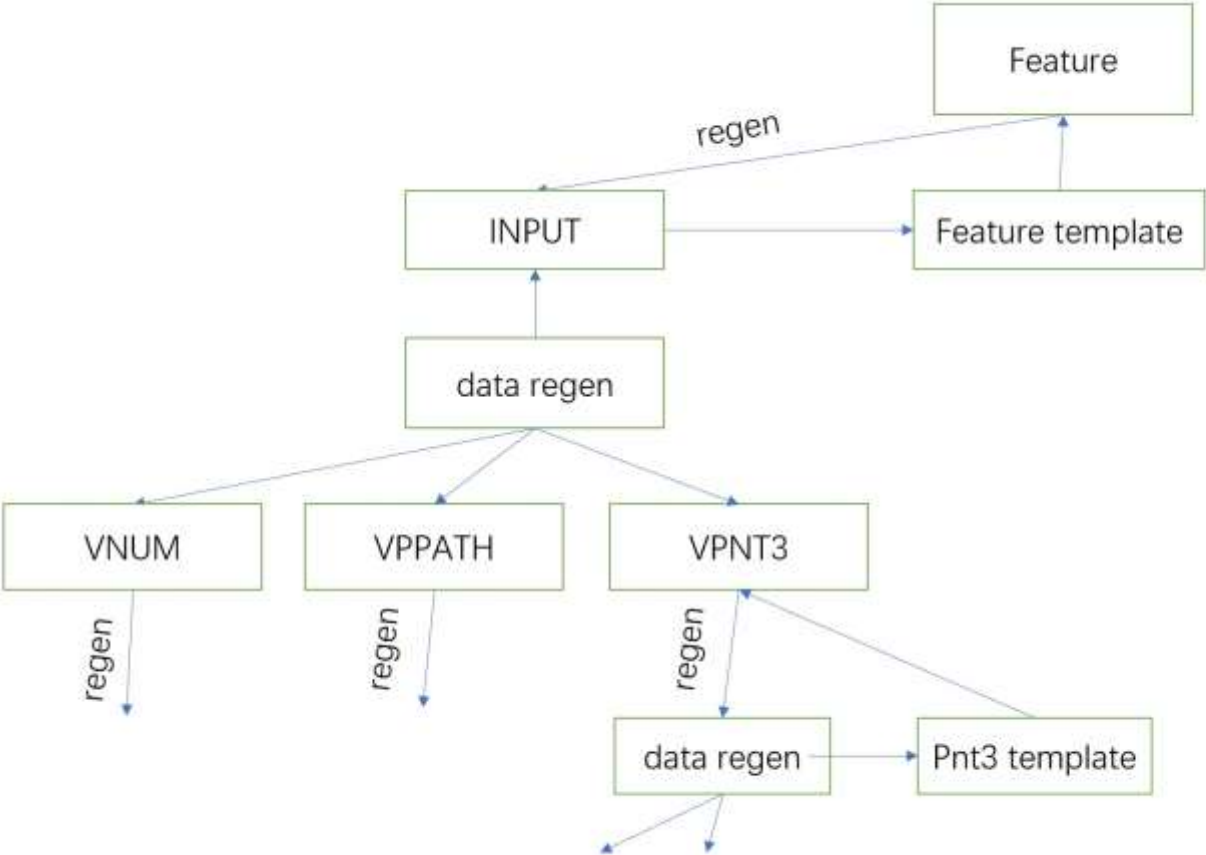
特征参数规则



```
CdNumPnt.cpp
392 int CdGetFaceset
393 {
394     int idx_in, /* I: Index of input data object (class=VDATA) */
395     int *idx_out, /* O: Index of output data object (class=VDATA) */
396     ...
397 }
398 /*
399 DESCRIPTION:
400 Create a VNUM object in the PRJ_BIN whose value is zero, with the
401 input points (on a boundary faces) object attached to it.
402 Output the index of the new VNUM object. The index will be piped
403 to the active field of the active form.
404 ...Field 1-- Points (on faces) pick
405
406 Return 1 if error, else 0.
407 */
408 {
409     ...int iCnt = 0;
410     ...if (VxInpCnt(idx_in, 1, &iCnt) || iCnt == 0)
411     {
412         ...return 1;
413     }
414     ...if (VgFtrRegen == 0)
415     {
416         ...return 0;
417     }
418     ...
419     ...VsojHandle oh, toh;
420     ...C_AUTO_REVERT_BIN();
421     ...CdDispCommand();
422     ...
423     .../* get a copy of the point pick object */
424     ...{
425         ...OnInitOhByCurBin(&toh, -1);
426         ...int idx_toh = toh.getIndex();
427     }
428 }
```

符合基本
应该有

特征参数规则

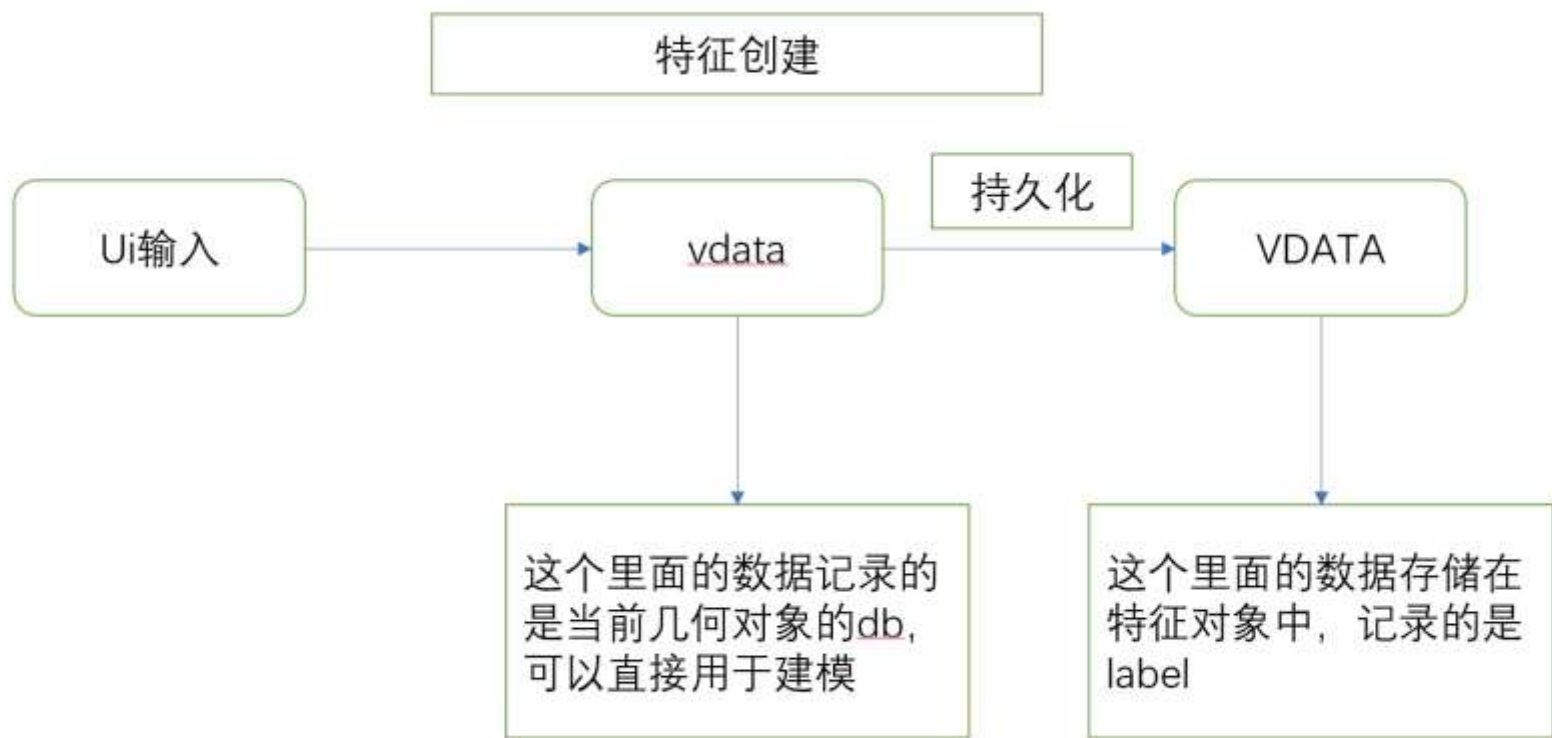


特征参数规则

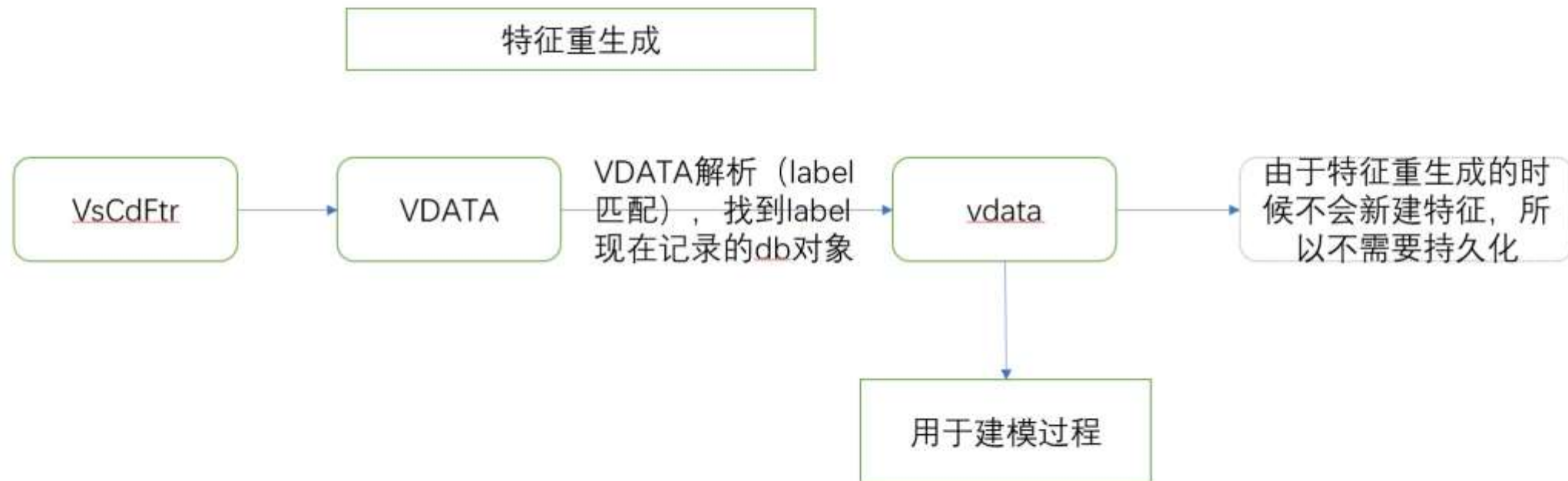
- 1、ZW3D的模版功能强大，基本可以适配各种功能，不要为了做到各种奇怪的功能加太多逻辑
- 2、鼓励大家使用自定义模版，当然是正确的使用，我们会整理出已有的模版，来达到复用的效果

V DATA的使用规范





VDATA



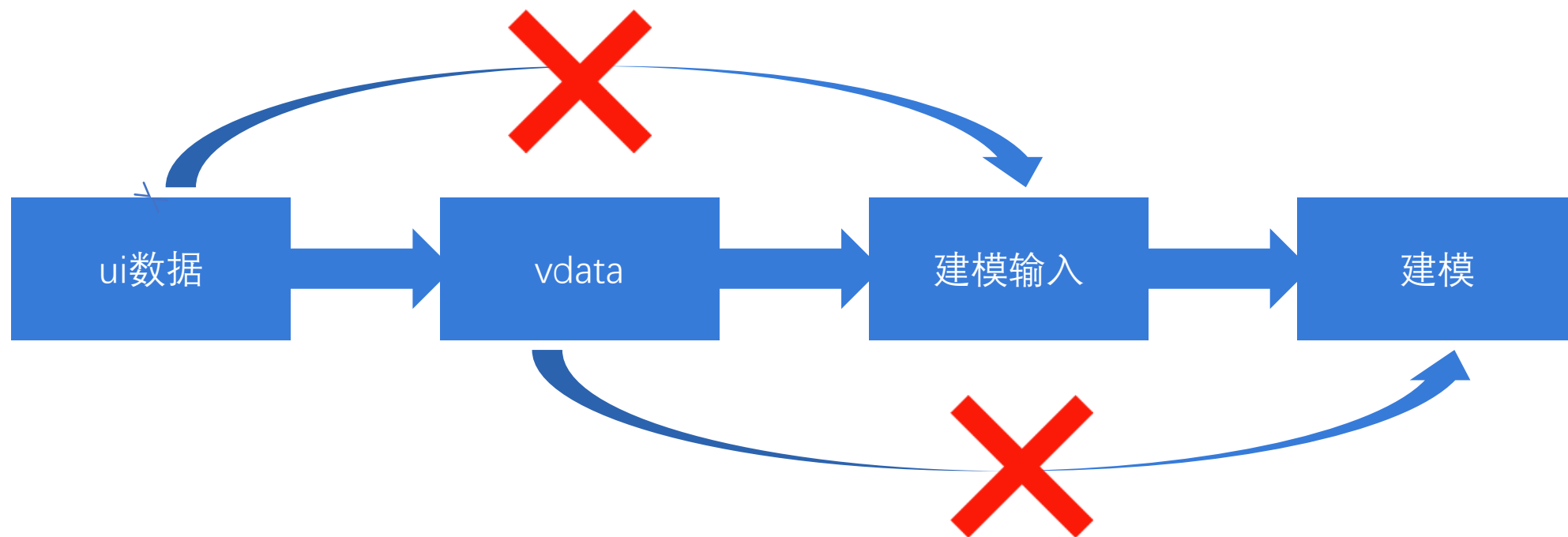
VDATA

```
.../*Get the boolean state*/  
} ...if (VxInpInt(idx_in, 14, &iCombine))  
...{  
...    iCombine = VxSymCmdIntGet(VxDataName(idx_in), 14, V_FTR_MRGE);  
...    VxLogNum(idx_in, 14, V_NUM, iCombine);  
...}  
...VxSymCmdIntSet(VxDataName(idx_in), 14, iCombine);
```

```
/*Create the surface*/  
try  
...{  
...    FtBridgeSrfBase(V_FALSE, idx_in, phEcho);  
...    VxInpDel(idx_in, V_FLD_FACE_START);  
...    VxInpDel(idx_in, V_FLD_FACE_END);  
...}
```


VDATA

获取类型	接口	备注
输入项	VxInpIdx、VxInpList、VxInpItem	
Pick path	VxInpPathOnly、VxInpPathList	不修改src_bin
实体	VxInpOh、VxInpEnt、VxInpEntList、VxInpTtEnts、VxInpTtRef	会修改src_bin
数值	VxInpInt、VxInpNum、VxInpAng、VxInpDst VxInputInt、VxInputNum、VxInputDst、VxInputAng VxInputNewInt、VxInputNewNum、VxInputNewDst、VxInputNewAng	旧式 新式 新字段
点	VxInpPnt、VxInpPntList	
方向	VxInpDir、VxInpDirList、VxInpIdxDir、VxInpEntDir	
其他	VxInpPrfSingle、VxInpGen、VxInpClass、VxInpStr、VxInpBox、 VxInpFtr	



感谢聆听

Q&A