

ZWsoft

装配一键重生成

2024/9/6

CAD平台开发部/装配设计组/装配平台组——Marta

装配关联更新是指在产品设计过程中，当源文件（如零部件设计文件）发生变化时，与之相关的下游设计数据（如装配体、工程图等）能够自动或手动地进行更新，以保持设计的一致性和最新状态。这种更新机制对于提高设计效率、减少错误以及加快产品上市时间具有重要意义。

装配关联更新的实现依赖于软件的技术支持和正确的操作流程。例如，在SolidWorks中，通过其关联机制，可以在短时间内教会用户如何理解和应用装配体的关联性。在Inventor中，当模型导入后，如果源文件发生设计变更，Inventor会自动检测并提示更新，用户只需进行模型更新，文件便会自动重新导入，同时用户在Inventor中所创建的下游设计数据也会一并更新。此外，装配体文件内存储的技术使得关联更新的设计数据易于管理，便于修改和储存。

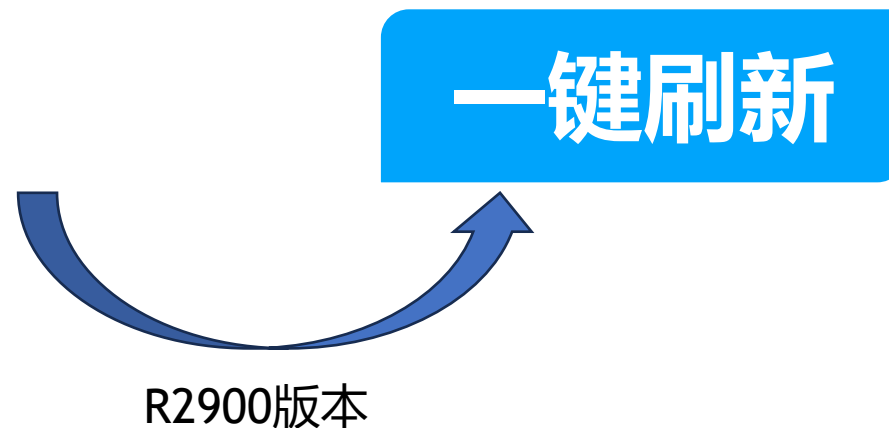
在Creo中，通过使用合并/集成指令，可以确保后续加工制造的零件模型和产品模型是分开的，同时又能保证后续模型能够关联更新，及时获取产品零件的最新变化。这种技术不仅提高了设计的灵活性，也保证了设计的准确性。

装配关联更新的另一个重要方面是在装配体中的配置管理。通过正确的配置映射和应用，可以确保零部件之间的关联关系随着配置的变化而更新。例如，在子装配体内的零件与顶层装配体之间的关联关系，需要通过正确的配置映射和应用来保持同步更新。

综上所述，装配关联更新是一种设计过程中的重要技术，它通过自动或手动的方式更新设计数据，以保持设计的最新状态和一致性，从而提高设计效率和产品质量 1 2 。

[收起 ^](#)

一键重生？



你能了解到什么？

装配重生成

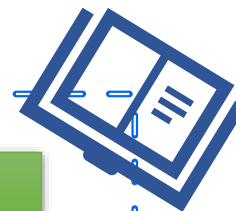
- 当前装配重生成的几种方式
- 与零件重生成的区别
- 哪些对象都参与装配重生成

链接管理器

- 链接关系的类型有哪些
- 应用场景有哪些
- 链接关系的构建方式

一键刷新

- 排序方式
- 重生成方式
- 多配置下如何重生成
- 待优化的点

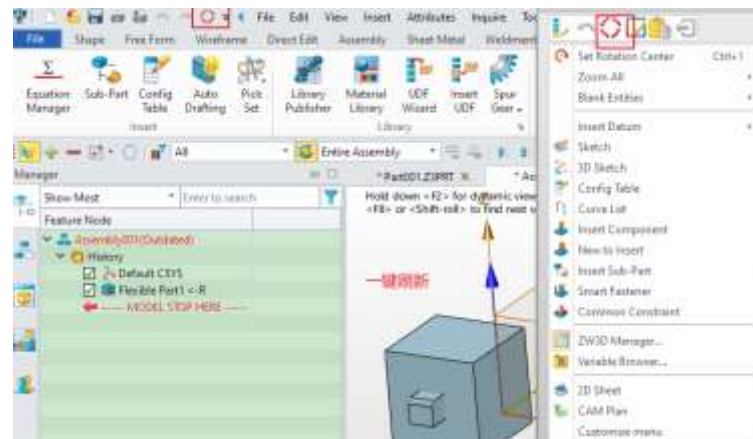
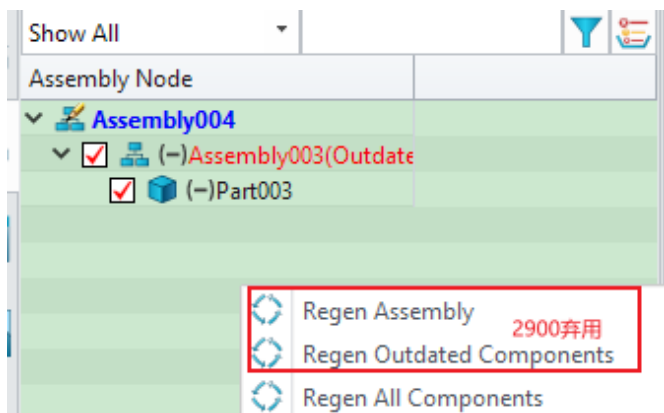


目录

C O N T E N T

- 一. 装配重生成
- 二. 链接管理器
- 三. 装配一键重生成
- 四. 相关文档

➤ 装配重生成的方式



➤ 与零件重生成的区别

- ① 重生成范围不同
- ② 装配重生成依赖特征依赖和链接关系，零件不依赖
- ③ 零件不需要排序，装配需要
- ④ 零件重生成一次，装配如果有过时会迭代多次

➤ 哪些对象会参与重生成



目录

C O N T E N T

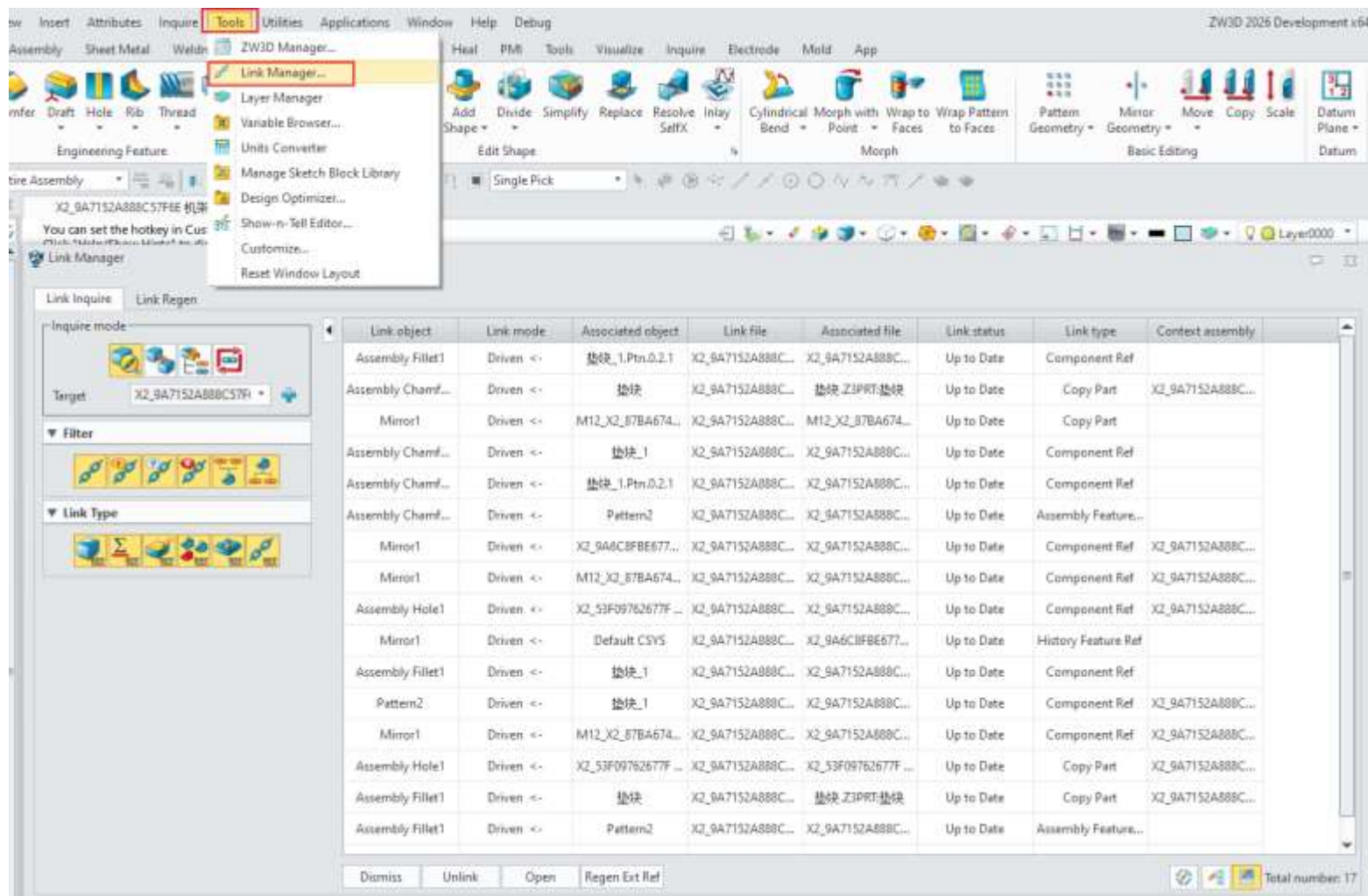
一. 装配重生成

二. 链接管理器

三. 装配一键重生成

四. 相关文档

- ◆ ZW3D中的链接管理器是用来管理并呈现零件内部以及零件与零件之间的引用关系的



➤ 常见的数据引用类型

◆ ZW3D中装配数据的引用形式

变量表达式引用

- 变量与变量
- Dim与变量
- 条件抑制
- 用户属性

实体参考引用

- 草图内参考(CdPrfRefxx)
- 参考特征(CdPartRefxx)
- 发布集引用(CdPartPubSet)
- 特征/约束实体引用(隐式)

草图引用

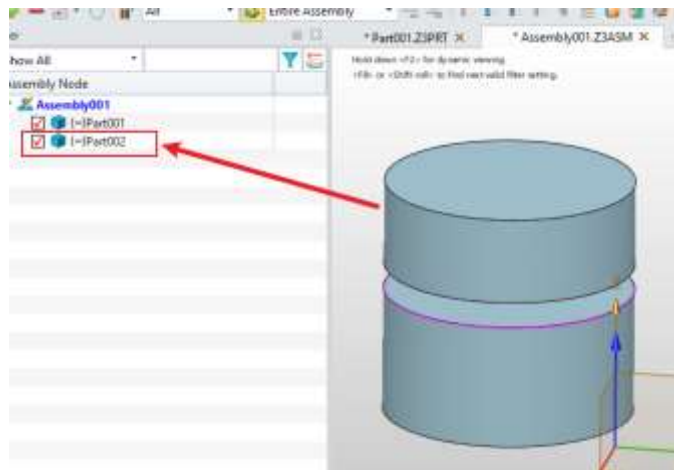
- 插入草图(CdInsSketch)

组件/零件引用

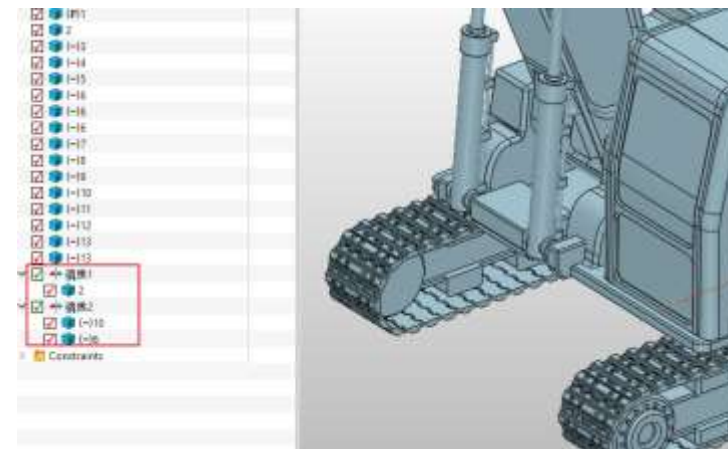
- 装配特征
- 装配上下文特征
- 插入零件特征(CdInsPart)



表达式引用



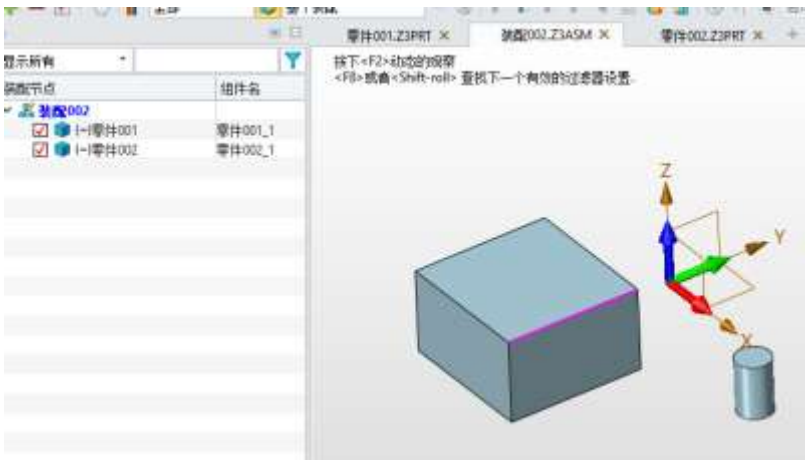
实体引用



组件引用

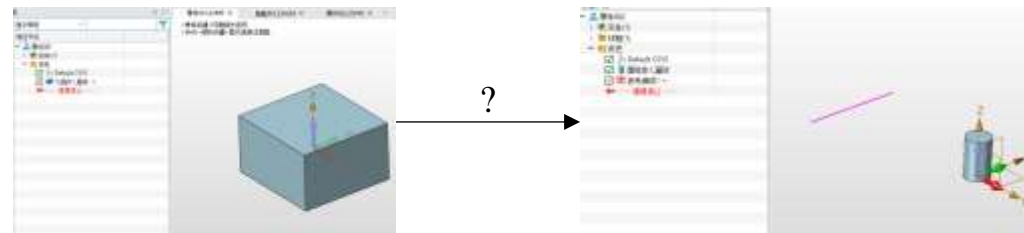
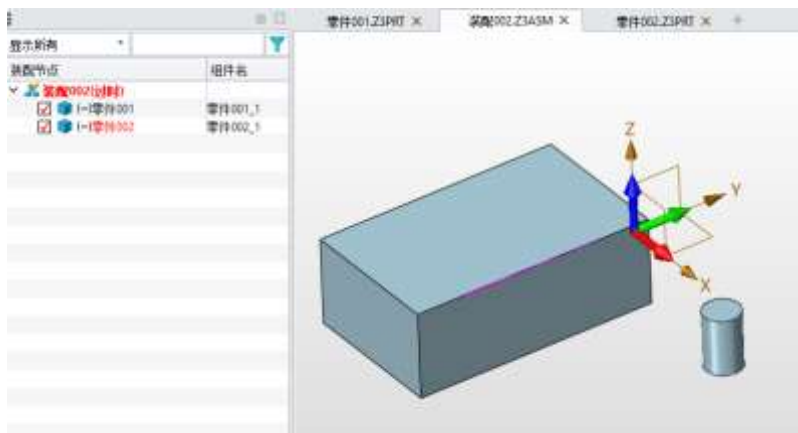
案例分析

- ◆ [案例]: 装配002中的组件[零件002]参考了[零件001]中特征产生的一条边。



(简化的模型)

Q1: 当重定义 [零件001]中基体特征怎么通知[零件002]的参考曲线特征过时?



Q2: 我们该建立怎样的链接关系?

边



参考曲线

➤ 链接管理器 (Link Manager)

1.链接管理器的定义:

运行时动态管理和维护同文件、跨文件之间的数据引用关系，用以通知过时、关联更新等。

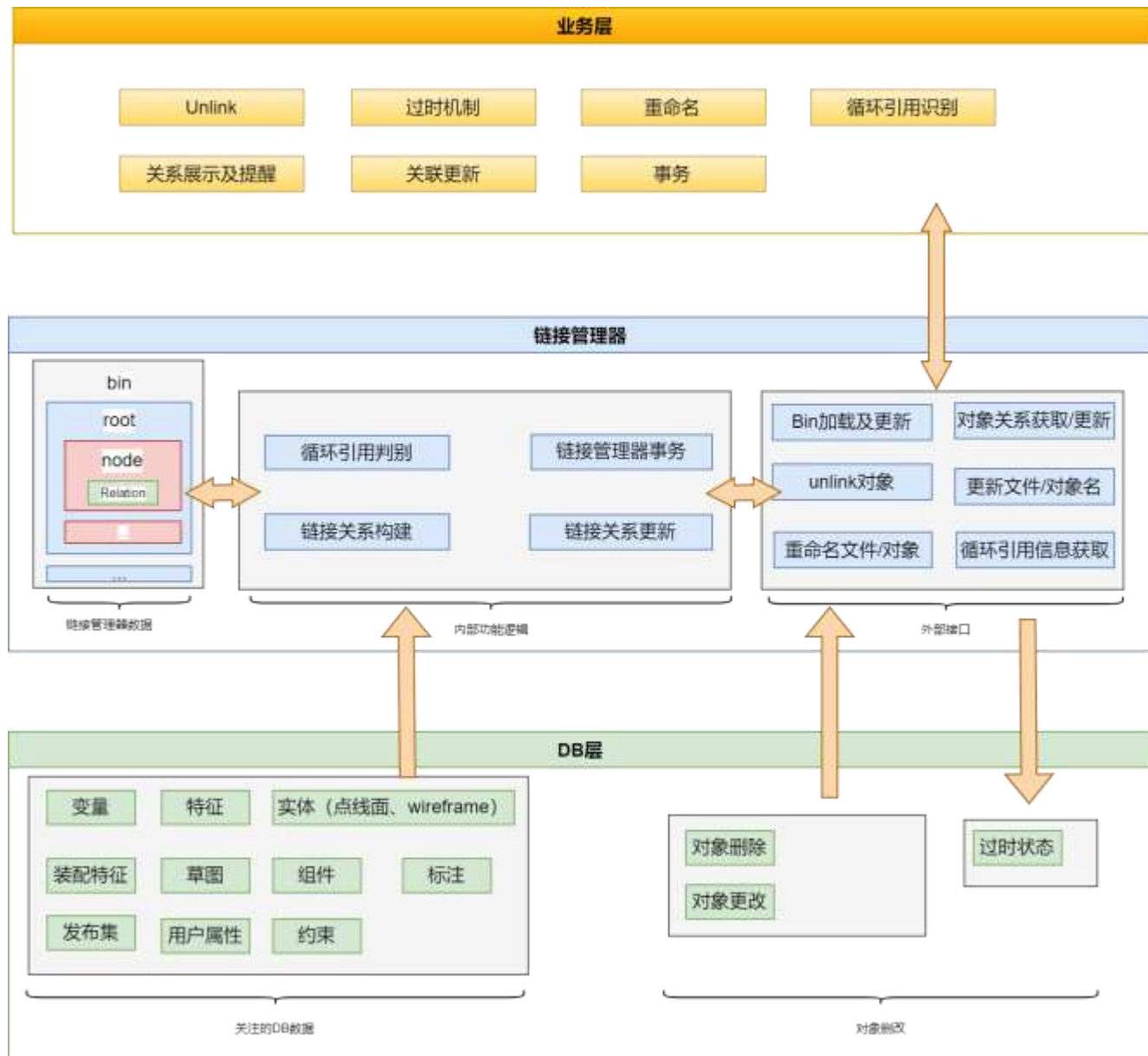
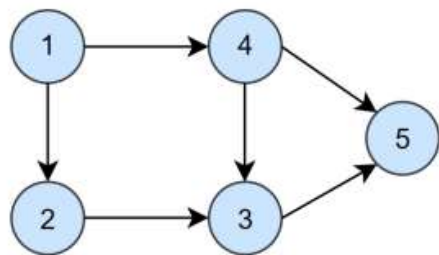
2.链接管理器的数据流:

作为业务层与数据库的中间层，数据流向如右图所示。

3.链接关系的数据结构:

数据具有指向性，将所有链接关系抽象成一个庞大的图，建立有向无环图的拓扑结构用来维护着节点和边。

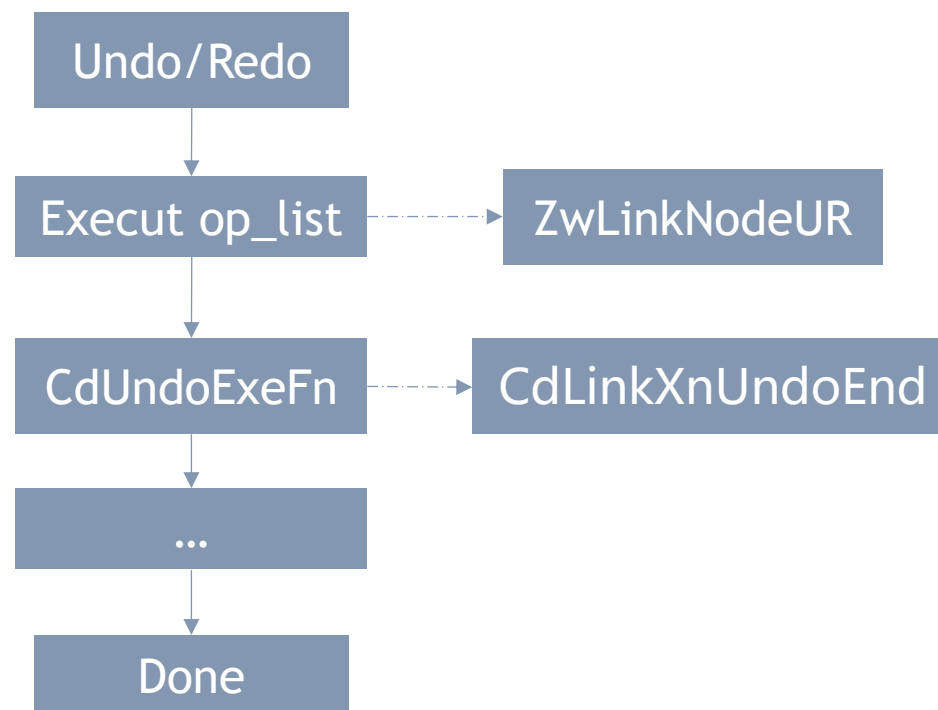
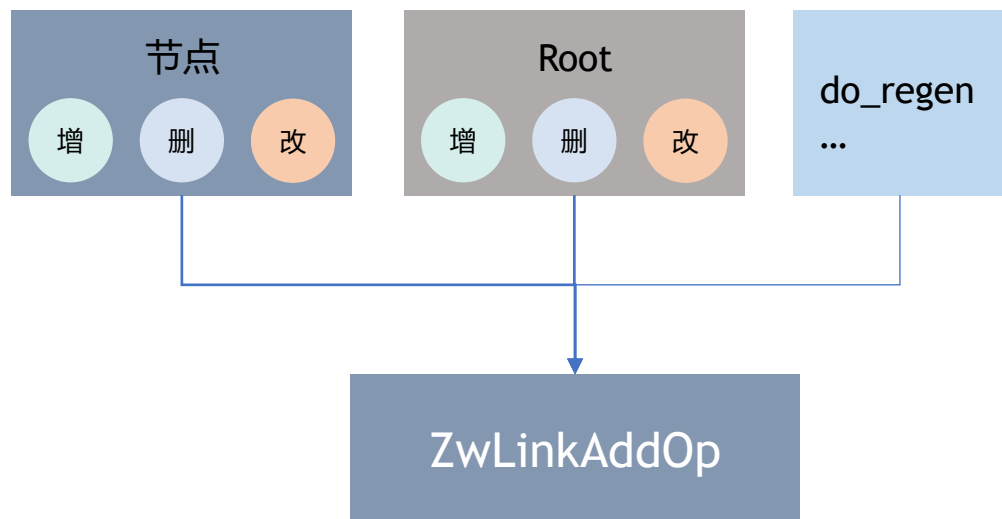
(DAG: Directed Acyclic Graph)



➤ 链接管理器 (Link Manager)

4.链接管理器事务:

1. 通过ZwLinkXnAddOp记录节点信息 (Bin+Root+其他信息) 注册在平台事务内, 利用平台的事务回调机制去动态维护链接管理器中的数据。
2. Undo, Redo时触发事务级别的回调函数CdLinkXnUndoEnd, 再按照顺序执行ZwLinkTransaction::ZwLinkUR(), 根据事务类型进行分发处理。

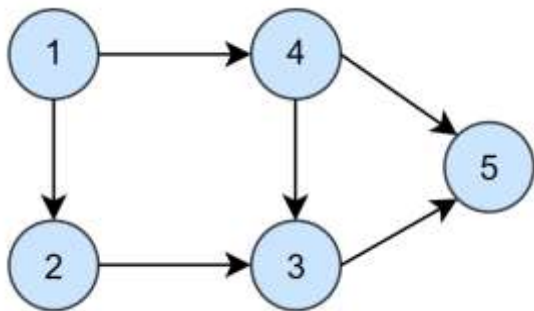


➤ 链接管理器 (Link Manager)

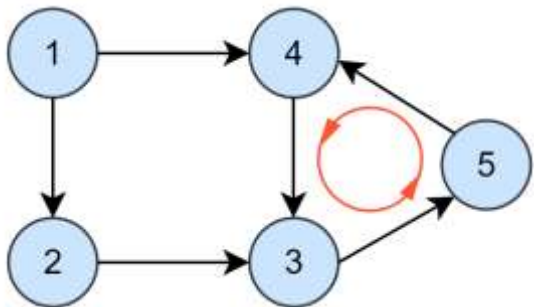
5.循环引用识别:

1. 循环引用: 有环的有向图, 不能进行拓扑排序。关于循环引用的识别: [基于混和图循环引用识别](#)

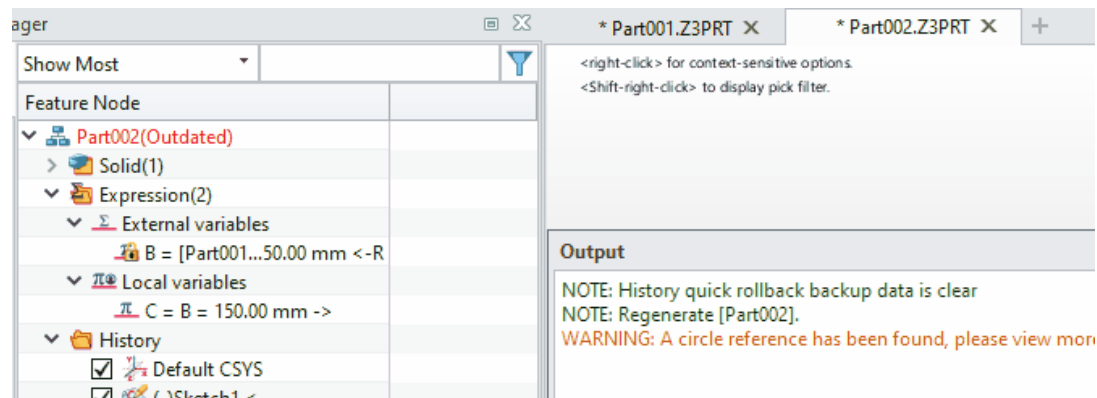
2. 案例:



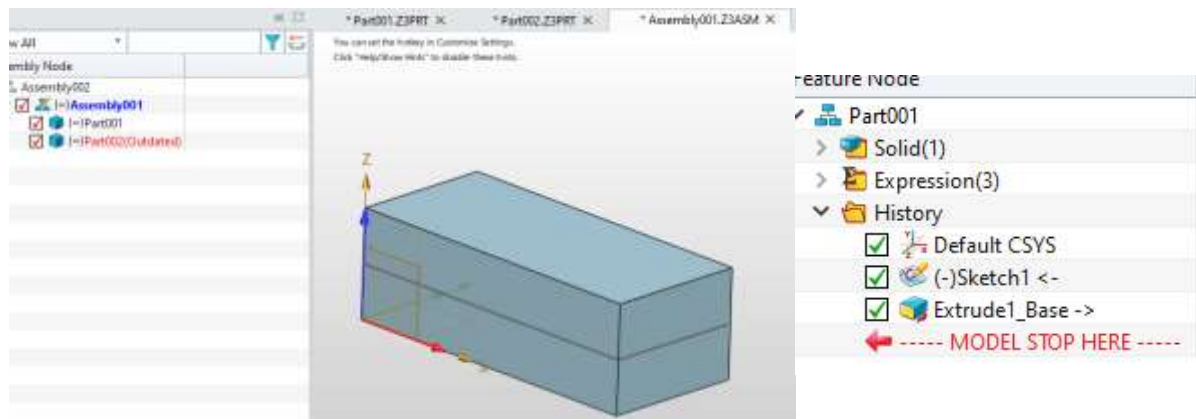
DAG



具有循环引用的有向图



变量循环引用



历史时序循环引用

➤ 其他

◆ 链接管理器标记



- 特征(R) (X): R代表过时、X代表Missing;
- (<-) (->): 分别代表被驱动(target)、驱动源(source);
- (Outdated标记): 表示零件过时(do_regen), 历史过时(1)-装配特征过时(2)-约束过时(4)-变量过时(8)

➤ 其他

◆ 调试小技巧

➤ 链接关系构建

- ✍ 文件打开时: ZwLinkBinOpen->ZwLinkRoot::loadRoot->loadRootCallBack->ZwLinkBuilder::proc_** (处理不同对象的解析)
- ✍ 命令执行结束 (新建、重定义、重生成) :FtrPostEval->CdLinkNodeUpdate->freshNode->loadRootCallBack->ZwLinkBuilder::proc_** (处理不同对象的解析)
- ✍ 其他具体业务流程中, 例如标注引用表达式: CdDimFtrModVal->CdLinkMgrPartDimUpdate->freshNode->***

➤ 设置过时

- ✍ VsCdPart->do_regen; //条件断点
- ✍ ZwLinkRelation.setOutDated(); //链接关系设置过时
- ✍ CdPartSetDoRegen (); //链接管理器设置Part中do_regen的唯一接口 **不要使用CdObjVarSet设置do_regen字段**

目录

C O N T E N T

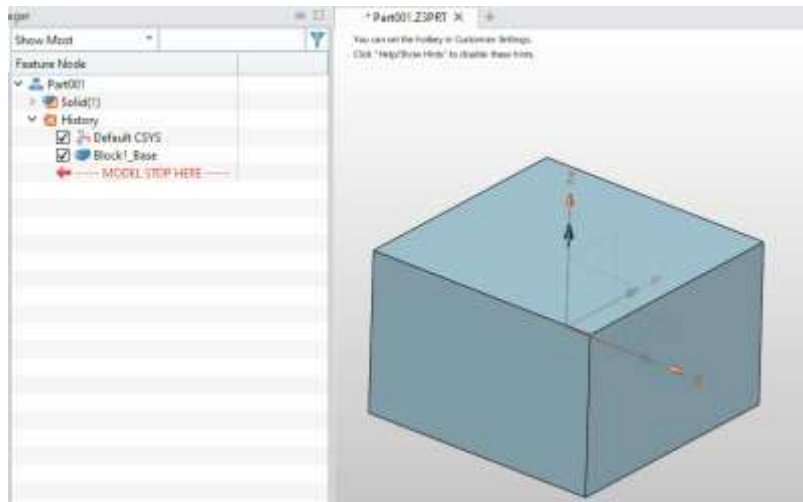
一. 装配重生成

二. 链接管理器

三. 装配一键重生成

四. 相关文档

➤ 装配更新的变与不变



(无参数改动的特征重定义)

◆ 不变：

- 1.零件、特征与变量的数据库索引；
- 2.VDATA记录的label；

...

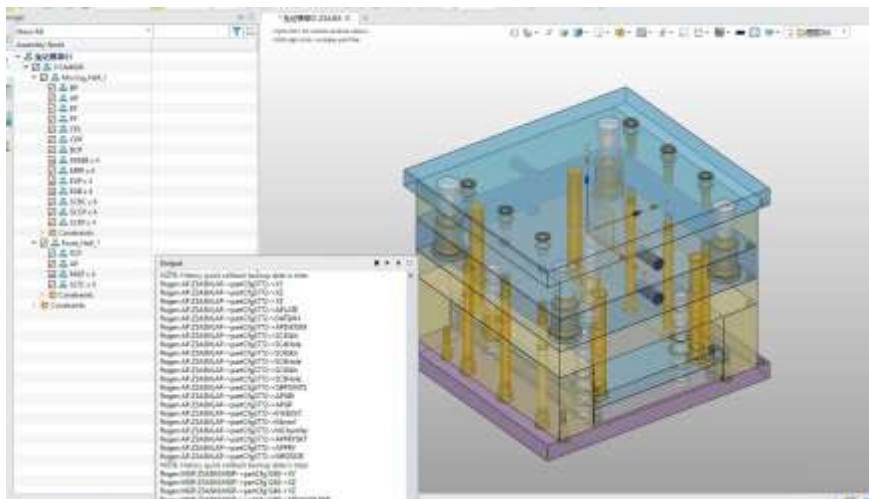
◆ 变：

- 1.特征重生成出的实体、标注；
- 2.装配特征生成的组件；

...

装配重生成：以单个历史树节点重生成为基础，以参数为驱动，进行整个装配模型的更新

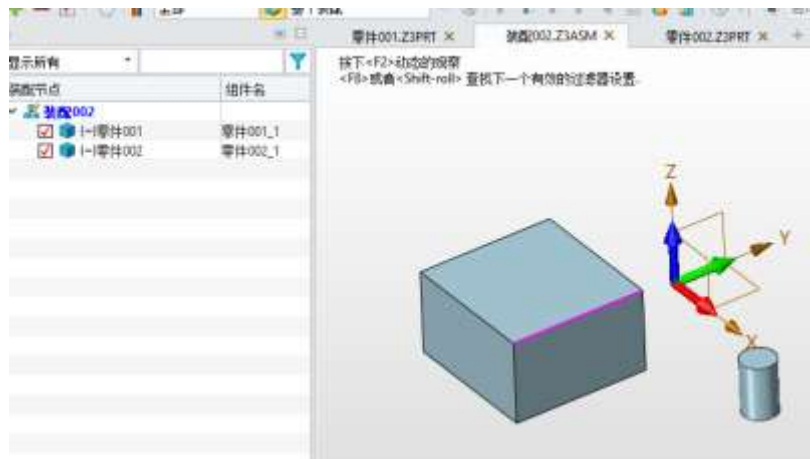
备注：PA新机制后，实体、标注也是不变对象，
可变的对象有组件，copy的草图



(无参数改动的装配重生成)

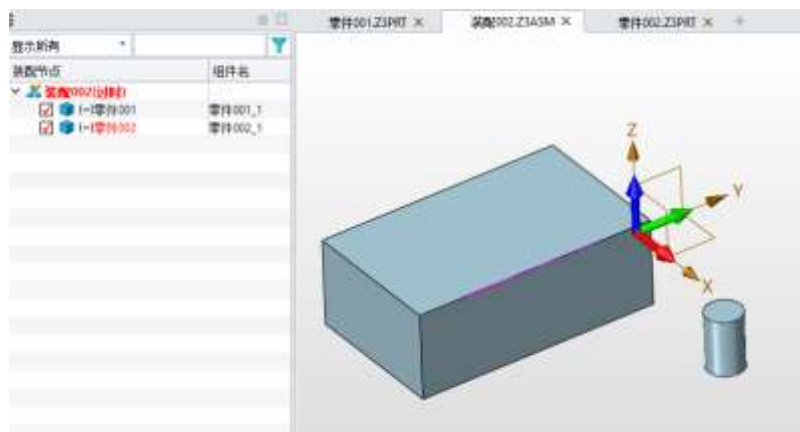
➤ 关联更新案例分析

- ◆ [案例1]: 装配002中的组件[零件002]参考了[零件001]中特征产生的一条边。



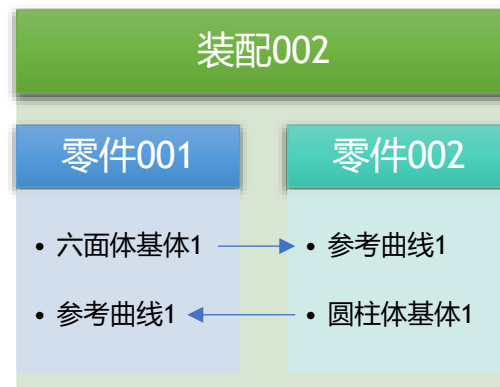
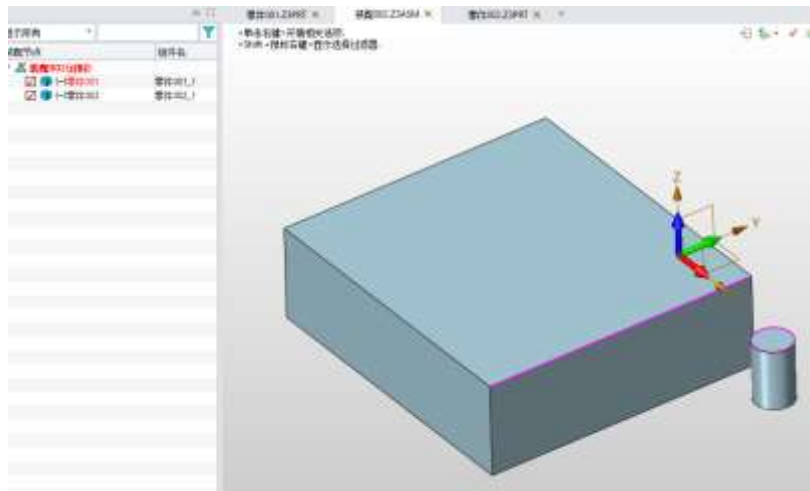
(简化的模型)

Q1: [零件001]、[零件002]都过时, 怎么从装配的角度重生成这两个零件?



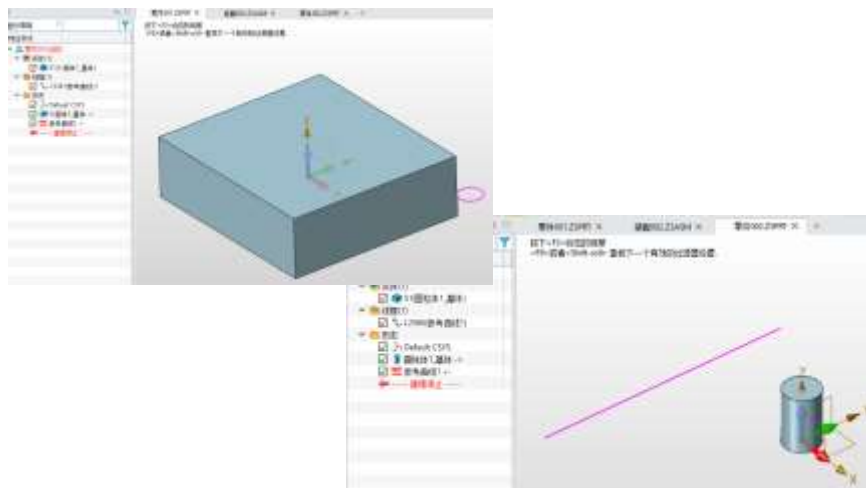
➤ 关联更新案例分析

- ◆ [案例2]: 装配002中的组件[零件002]参考了[零件001]中特征产生的一条边, [零件001]也参考了[零件002]特征的一个边。

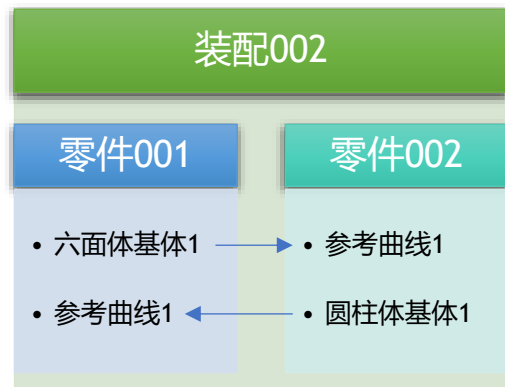


(简化的模型)

Q2: [零件001]、[零件002]都过时, 怎么从装配的角度重生成这两个零件?



➤ 装配重生成的方式



(简化的模型)

◆ 方式1：以零件为颗粒度重生成

- 优点：将零件作为一个整体重生成，易与理解、调试，在零件之间数据无关联时可利用并行提高效率。
- 缺点：零件级循环引用非常普遍，需要反复迭代同时利用真实过时更新，有时无法去除过时；



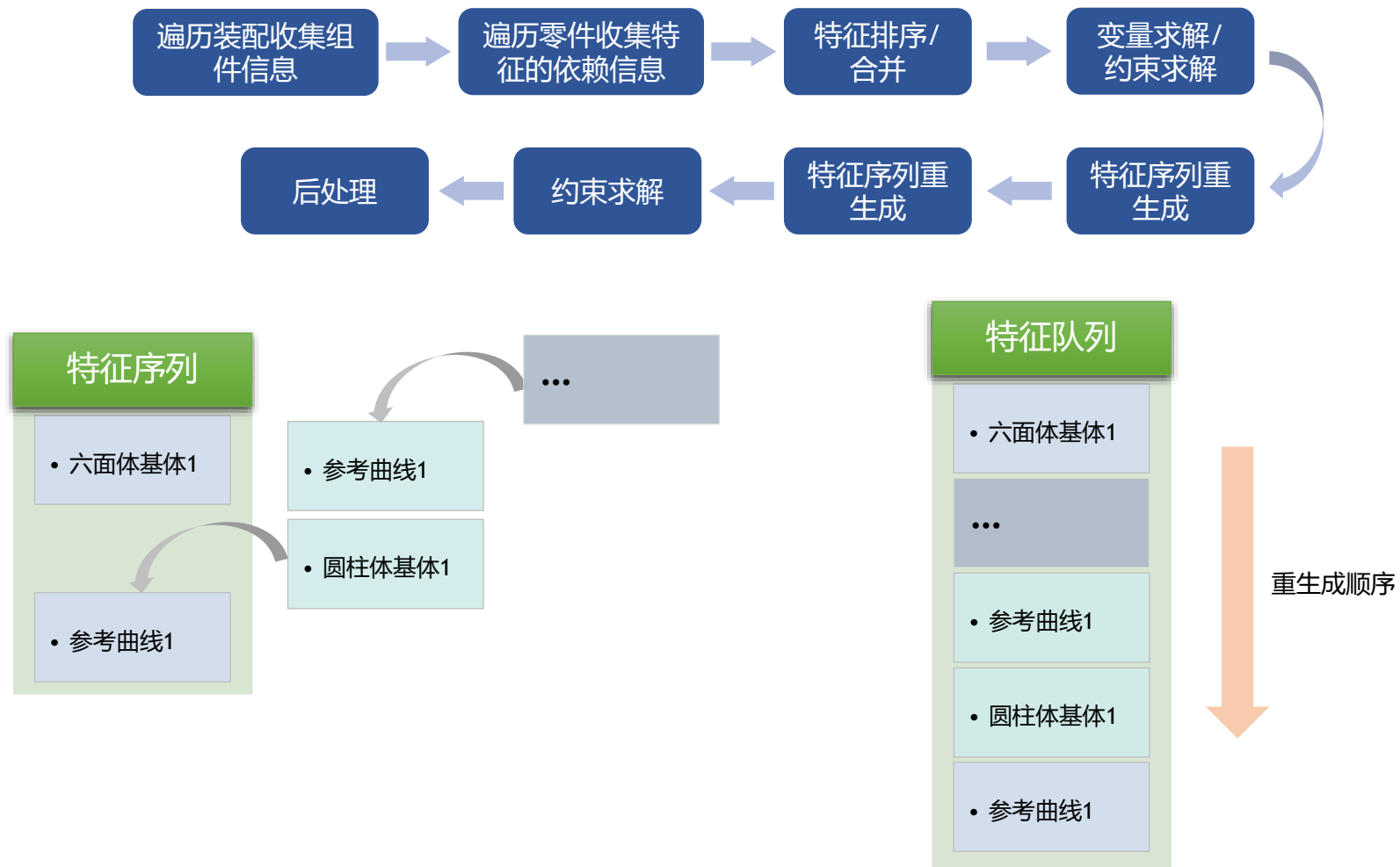
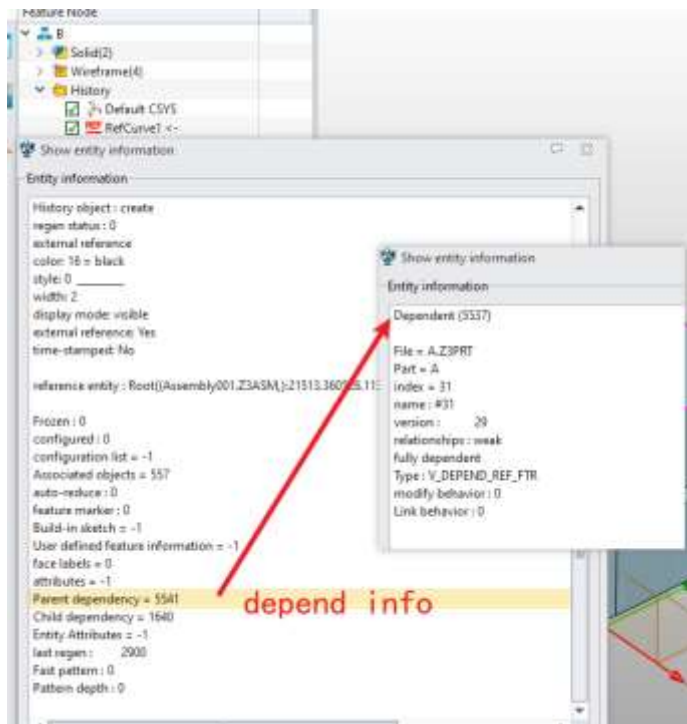
◆ 方式2：以特征为颗粒度重生成

- 优点：解决零件级的特征循环引用，减少重生成迭代。
- 缺点：需要保证依赖信息的准确性，数据处理/调试复杂；
- \$PrintAsmRegenInfo=1



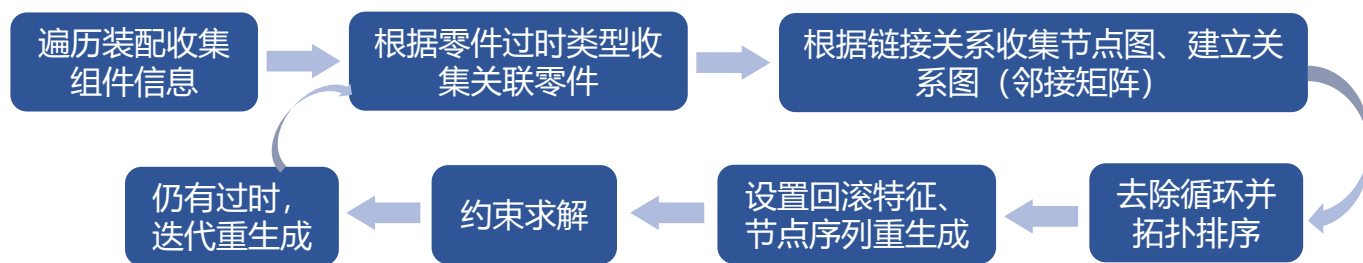
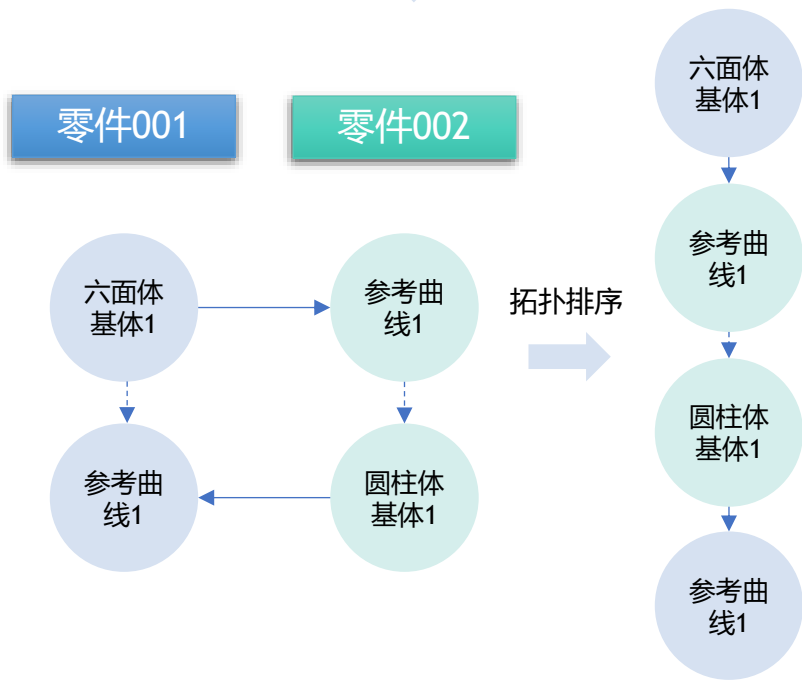
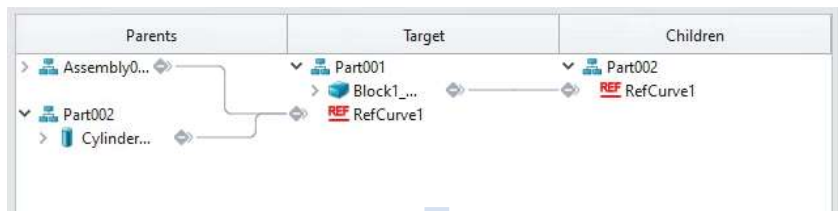
➤ 特征序列的再生成（重生成所有组件）

- ◆ 利用特征记录的depend info排序，详见Wiki页面：[特征序列的装配重生成](#)



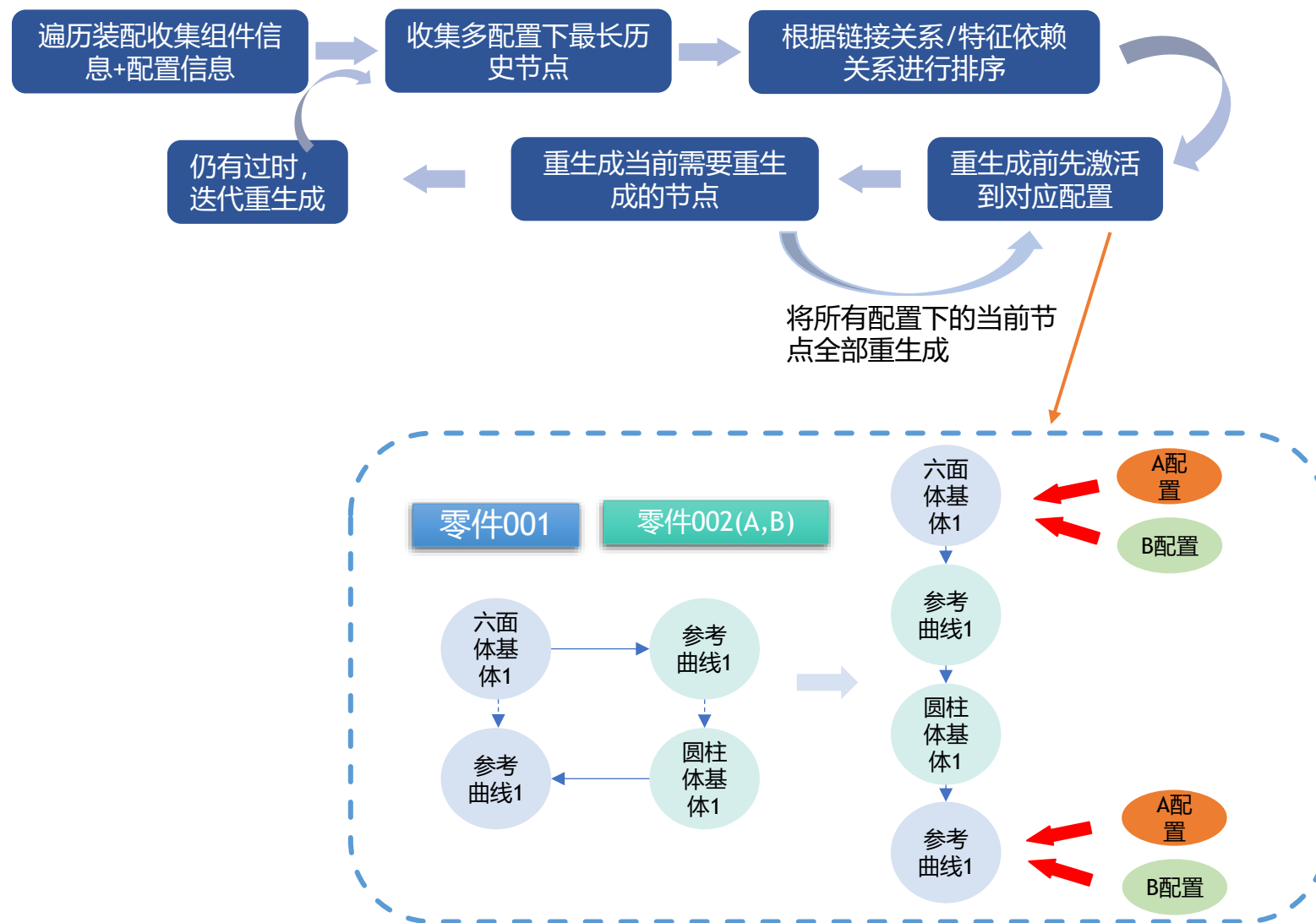
➤ 项目改进（一键刷新）

- ◆ 基于链接关系的一键重生成，详见Wiki页面：[基于链接关系的一键重生成](#)



- 将变量考虑进来参与排序，减少重生成次数；根据链接类型收集过时零件，减少参与重生成的零件，同时利用回滚特征的方式进行重生成，提高了重生成的效率；
- 进一步提高了对链接关系的准确度要求；
- 约束求解需要与特征重生成穿插进行的用户场景只能通过迭代方式解决；
- 对于外部修改的零件导致的过时，将来需建立一套校验的机制

➤ 多配置重生成



◆ 重生成所有与一键刷新对比：

	重生成所有	一键刷新
重生成范围	装配下的所有组件	只有过时的组件
排序依赖	特征依赖关系	链接关系+特征依赖
参与排序的对象	历史特征、装配特征	历史特征、装配特征、表达式、pubset
历史特征重生成范围	从第一个节点开始	标记过时后，采用回滚的方式从过时特征开始
失败处理	Undo重生成操作，还原到重生成之前状态	继续向下重生成
过时处理	迭代3次	迭代5次

◆ 待提升的点



目录

C O N T E N T

- 一. 装配重生成
- 二. 链接管理器
- 三. 装配一键重生成
- 四. 相关文档**

➤ 装配重生成专项

- ① 链接管理器重构
- ② 基于特征序列的一键重生成
- ③ 基于链接管理器的一键重生成



- ✓ **重生成所有组件**
- ✓ **一键刷新**

➤ 装配设计

Q&A

• THANKS

感谢观看