基于组件的机械系统虚拟样机可视化建模研究

黎国华[1]，熊会元[2]

(1. 武汉理工大学，湖北 武汉 430070; 2. 中山大学 工学院，广东 广州 510275)

摘　要: 应用组件技术,将机械系统虚拟样机分析系统设计为独立于几何模型的组件,这样简化了系统分析的复杂性,提高了可扩展性和可移植性。采用基于组件技术的三维引擎 ACIS,重用其基本功能组件,实现了机械系统虚拟样机的几何构件体,约束和载荷的可视化建模。

关键词: 虚拟样机; 组件; 可视化建模

中图法分类号: TH113; TP391　　**文献标识码**: A　　**文章编号**: 1001-3695(2004)06-0073-03

Research on Visible Modeling of Mechanical System Virtual Prototype Based on Components

LI Guo-hua[1], XIONG Hui-yuan[2]

(1. *Wuhan University of Technology, Wuhan Hubei* 430070, *China*; 2. *School of Technology, SUN YAT-SEN University, Guangzhou Guangdong* 510275, *China*)

Abstract: Design a virtual prototype analyze system as a separate component apart from geometry model with application of components technology, which simplifies the complexity and improves the scalability and transplantability of analysis system. Develop relevant arithmetic, which realize visual modeling about geometry parts, restrictions and load of a virtual prototype analysis system and reusing basic functions components of the ACIS, a 3D-geometry model engine based on components technology.

Key words: Virtual Prototype; Component Technology; Visual Modeling

目前,国内虚拟样机设计分析软件市场基本上被国外产品所垄断,较有影响的有 ADAMS, DADS, SIMPACK 等。这些软件没有中文版,价格较高,对于国内的大多数企业来说可望而不可及。开发虚拟样机分析系统是一项技术要求高、投资大、周期长的工程。这样，如何重用已有的研究成果和将自己的研究成果开发为可重用包，这对可持续的研究开发显得十分重要。本文应用多体系统动力学基本原理，组件技术，采用面向对象的分析与设计方法,开发了一个与几何模型分离的虚拟样机分析系统组件 MASS，并在开发的原型系统中得到了应用。

1　基于组件的虚拟样机系统

机械系统虚拟样机集成了自动建模、微分代数求解、数据管理和仿真等技术,是一个庞大而复杂的系统,传统的软件设计的方法,不能很好地适应开放式虚拟样机系统实现的要求。组件技术的研究应用和逐渐成熟为解决这一问题提供了坚实的基础。基于组件的系统是由若干个具有标准接口的可重用独立模块组成,接口是组件间进行通信的唯一途径。每个组件都是经过编译链接好的二进制代码;组件强调组件模块内部的封装性,其内部的实现是隐藏的,对应用来说是透明的,是实现对象级重用;它具有与语言无关、进程透明和可重用性等特征。应用组件开发的复杂系统具有很好的开放性,可自由拼装形成不同的功能配置,因而备受青睐。

收稿日期: 2003-02-17; 修返日期: 2003-07-16

1.1　机械系统虚拟样机系统结构

应用组件开发系统的主要内容是根据应用系统的功能需求列出所有构成组件、各个组件间的依赖关系和接口,并确定哪些组件自己开发、哪些重用、哪些直接购买,以缩短开发周期。本文应用基于组件技术开发的三维造型引擎 ACIS 的基本功能组件,重用本中心开发的基础组件作为辅助显示管理工具,而集中开发核心组件 MASS。系统结构如图 1 所示。

ACIS 核心组件包是美国 STI 公司开发的,完全基于组件技术的三维几何引擎,提供几何造型、视图操作、文件管理、内存管理等基本功能组件。CBA 是自行开发的基本数学运算和几何操作组件,提供堆、链表、队列、矩阵、向量等的定义与操作。BASEGI 是从 ACIS 相关组件中派生,用于几何图形的显示、刷新等操作。3DTOOL 提供几何造型对话框、鼠标操作等的定义。BASEPART 定义各类几何造型、约束几何模型的超类,同时也为模型管理提供超类。DEFRB 定义几何建模过程中显示橡皮条的超类。NAVIGATE 为几何造型、添加、约束和编辑模型等操作提供导航服务组件。XTREE 是自行开发的一个可视化模型管理树组件,为系统提供动态模型管理信息显示。MASS 是本系统的核心组件,其封装了机械系统虚拟样机完整的类系结构。

1.2 核心组件 MASS 的设计

机构系统虚拟样机由可视化的几何模型、运动模型和求解的力学模型以及由此产生的数学模型组成的一个复杂的模型系统,每个模型又包含多种特征,具有多样性与复杂性。如何组织一种较好的模型储存结构和解决方案对整个分析系统至关重要。经分析,剥离可视化几何模型,抽象出模型的物理参数,建立一种与几何模型无关的模型数据结构(逻辑模型)。这样,一方面简化了问题的复杂性,另一方面清晰了系统开发结构。应用组件技术和面向对象的技术,采用 UML 辅助系统设计,应用笛卡儿建模方法,开发出基本的分析组件 MASS。图 2 为 MASS 的基本类系结构图。

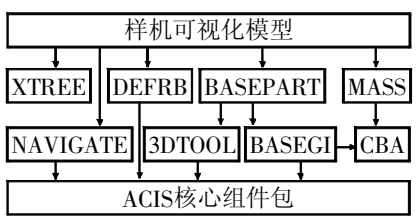

图 1 系统结构图

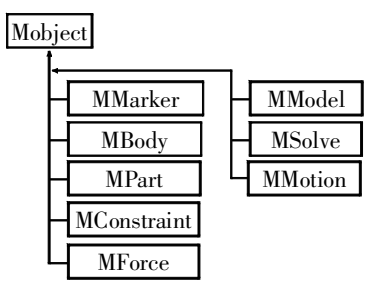

图 2 MASS 的抽象类结构图

(1) Mobject 系统的抽象类。包含对象名称、对象类型、对象 ID 和父对象指针等基本属性和对属性的操作,便于对象的分类与管理。

(2) MMarker 标记类。系统采用笛卡儿坐标,313 坐标变换建模方法。标记包含位置坐标(全局和局部的坐标)和姿态角(欧拉角)属性,提供方向余弦矩阵到欧拉角的转换。

(3) MBody 体类。体是组成构件体的单元,包含体的质心和特殊点的位置与姿态信息的标记对象和体的物性参数(惯性主轴,质量和惯性张量等)。

(4) MPart 构件体类。构件体是由一个或一个以上的体所组成的分析系统基本单元,提供体积、质量、惯性主轴和惯性张量等的组合运算。包含作用于体的所有外力或外力矩的载荷链表和确定位置与姿态的标记链表,计算作用于构件体上的合力矢和合力矩,以及力对广义坐标的导数,确定载荷与约束的作用点、方向等。

(5) MForce 载荷类。保存单个外力载荷,区别载荷类型,提供载荷属性的设置和访问操作。通过一组标记来记录表述载荷的作用点与方向,应用载荷函数定义载荷的特征。

(6) MConstraint 约束类。保存运动学约束的类型和定义数据,通过一组构件体上的标记来记录表述约束,同时计算对应约束的约束方程矢量、雅可比矩阵、速度右项矢量和加速度右项矢量。

(7) MMotion 运动类。保存每一时间步的位置,速度、加速度和拉格朗日乘子等运动物理参数,提供访问与设置操作。

(8) MModel 模型类。管理和储存整个模型,实现从力学模型到数学模型的自动建模,为求解器提供模型的数学模型,同时保存求解器求解的各种参数到运动链表中。

(9) MSolve 求解器类。获取模型类的数学模型,应用 ODAE 方法逐步迭代,迭代过程中将分析结果传送到模型类运动链表,并实时更新。

2 系统的可视化建模

MASS 中,模型是独立于几何模型的逻辑模型,工程应用中,须以可视化的交互方式来建立模型,这样结构逻辑模型和可视化模型给我们提出了要求。应用面向对象的分析设计方法,抽象出包含几何模型和逻辑模型的类系结构(在此称为运行时类系结构)。可视化建模主要包括体与构件体的建模、约束的建模以及载荷的建模。约束与载荷的可视化建模具有相似性,本文主要介绍构件体和约束的可视化建模。

2.1 构件体的建模

介绍构件的创建之前,首先介绍一下 ACIS 模型空间。ACIS 模型空间有一个默认的模型绝对坐标,称为全局坐标,同时在模型空间可以创建工作坐标,为了定位的需要,有时可以将工作坐标定义为当前活动坐标。本文建模方法采用 ACIS 的模型定位方法,抽象出一个坐标对象 CAcisMarker 类,来标志点的位置与姿态。

将几何模型与逻辑模型封装在一起,抽象出一个运行时的构件体类,一致性维护就变得简单。如下是抽象出体的超类,具体的实现类从此类派生。

```
class CAcisGBody: public CAcisObject    //构件体运行时超类
    {
private:
    MBody * m_pGBody; //逻辑体对象
    MPart* m_pPart; //逻辑构件对象
    BODY * m_pBody; //几何体对象
    CPtrList m_nMarkerList; //标记链表
public:
    virtual bool Create( ) =0;
    virtual bool InitGBody( ) =0;
    virtual bool InitGBodyOuLaAngle( ) =0;
    virtual bool InitGPartInertia( ) =0;
    virtual void GetMarkerCoordinateValue( );
    virtual bool Modify( ) =0;
    virtual bool ModifyGBody( ) =0;
    …
};
```

(1) 创建实现。从 3DTOOL 中的鼠标拾取超类派生出创建构件体的拾取类,参数化输入后,调用 InitGBody( ) 函数,该函数中,首先调用基本几何体创建函数 Create( ),利用 ACIS 提供的 API 函数创建基本的可视化三维模型对象,在创建几何体的同时,创建体上的质心点和其他特殊点的可视化标记,便于约束与载荷等的鼠标拾取。成功创建几何体,将其添加到模型管理多叉树中(图 3),接下来初始化 MASS 中的逻辑体对象,如果体为已存在的构件体组成部分,就直接添加到逻辑构件体体链表中,否则创建新的逻辑构件体对象,并将其加入到体链表中。初始化逻辑对象,调用 InitGPartInertia( ),赋值构件体物性参数,应用 ACIS 的 api_body_mass_pr( ) 函数获得体积、质量、体积惯性张量和惯性主轴等物性参数。随后,调用 InitGBodyOuLaAngle( ) 函数对构件体上标记位置与姿态赋值,在该函数中,通过 CAcisMarker 的 WCS 对象获得其模型空间坐标的位置和单位方向向量,将其转换为逻辑标记的方向余弦矩阵,进而计算出欧拉角。

(2) 修改实现。提供模型的修改是必要的。修改操作大致可以分为两种,即①整个构件体的几何特征符合要求,调整位置与姿态;②几何特征的修改。对前一种来说,主要是对对

象的平移和旋转操作; 后一种由于 ACIS 对生成的几何模型不能直接进行修改, 须将生成的对象删除, 根据新的参数重新生成, 这样, 其操作过程就如同创建过程。下面介绍第一种情况。

对模型修改必须获取待修改对象, 在本系统中, 提供利用鼠标直接从视图中获取对象和通过模型管理树获取对象的两种方法。主要介绍鼠标拾取的实现方法。应用拾取函数 Api _ gi _ pick _ entity( ), 获取视图中可视化实体对象指针, 要获得信息还必须找到封装它的运行时包容器对象。所有包容器储存在(图 3) 的一个多叉树中, 这样遍历多叉树来获取所需的包容器对象, 由鼠标获取的几何模型指针与多叉树中包容器对象中的几何模型指针匹配, 如匹配成功, 则得到了封装几何模型的包容器运行时类对象, 由包容器对象即可得到逻辑对象的指针, 这样就实现了对象的获取。

调用包容器对象中 ACIS 平移或旋转函数 TransformEntity ( ) 对几何构件体及其上的标记进行修改操作, 调用 InitGPartInertia( ) 和 InitGBodyOuLaAngle( ) 函数刷新逻辑构件体的位置与姿态以及所关联被修改对象的属性参数, 维护模型的一致性。

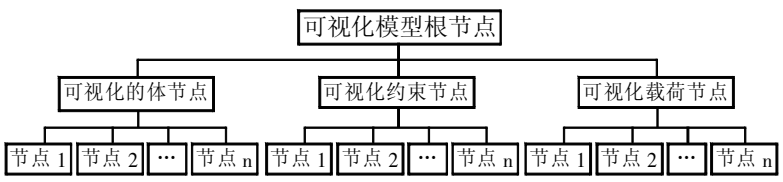

图 3　可视化模型储存结构图

2.2　约束建模

约束模型可以抽象为两个构件体间的关系操作, 就可视化建模来说, 就是获取约束的两邻接构件体, 创建约束可视化模型, 以及对具有方向特性的约束确定方向矢量。与构件体对象相似, 运行约束类中必须封装逻辑约束对象, 可视化约束示意图形和约束的方向向量等信息。

```
class CAcisGConstraint: public CAcisObject//约束超类
{
    private:
    MCconstraint *  m _ pGConstr; //逻辑约束对象
    CAcisMarker*  m _ pFirstMarker; //第一标记
    CAcisMarker*  m _ pSecondMarker; //第二标记
    WCS *  m _ pRefWcs; //约束位置
    BODY *  m _ pBody; //几何图形
    CAcisGBody *  m _ pFirstBody; //第一体
    CAcisGBody *  m _ pSecondBody; //第二体
    vector m _ nVector; //约束的方向向量
    public
    virtual bool AddMarkerToBody( ) ;
    virtual bool Create( )  = 0;
    virtual bool InitGConstraint( )  = 0;
    virtual bool InitGMarkerOuLaAngle( )  = 0;
    virtual void GetMarkerCoordinate Value( ) ;
    virtual bool Modify( )  = 0;
}
```

(1) 创建实现。拾取约束的两邻结构件体, 通过上述的匹配算法, 获得其包容器运行时类对象, 拾取约束点标记对象, 获得约束点, 应用 ACIS 的箭头橡皮条显示约束标记的方向向量, 记录上述所有对象, 调用 InitGConstraint( ) 函数, 该函数中, 首先执行创建约束可视化示意图和确定约束关系和方向的可视化标记对象的函数 Create( ), 创建成功, 则初始化逻辑约束的对象, 执行 InitGMarkerOuLaAngle( ) 函数, 对逻辑, 约束标记对象赋值, 提供 AddMarkerToBody( ) 函数添加可视化标记到相应的构件体对象, 并将约束对象添加到模型管理树中。

(2) 修改实现。同体与构件体的修改相似, 分为两种修改方式, 同样我们只对第一种情况进行介绍, 首先拾取需要修改的约束几何对象, 匹配多叉树获得封装可视化对象包容体运行时约束类对象, 调用 Modify( ) 函数修改, 与构件体修改不同的是, 平移旋转操作时, 只对约束关联的物理标记进行修改, 而约束几何图形不变。修改后, 调用 InitGMarkerOuLaAngle( ) 刷新逻辑标记的属性值和包容器类对象中关联的被修改的参数, 保证了模型的一致性。

3　应用实例

图 4 是应用开发的原型系统 inteVPAS 1. 0 建立的一个曲柄滑块机构模型, 该机构由曲柄连杆和滑块三个构件体组成, 包括曲柄与大地的转动副, 曲柄与大地的转动副, 曲柄与滑块的球副, 以及滑块与大地的弹簧约束。在曲柄与大地转动副上加载单作用力矩, 滑块与大地加载单作用力和弹性力, 以及所有的构件体所受的重力。求解结果与 inteVPAS 1. 0 生成的. adm 文件在 ADAMS 上的求解结果相符。

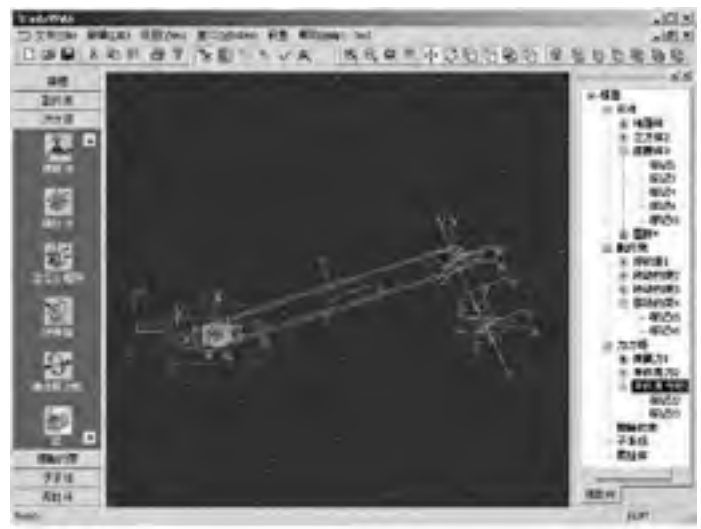

图 4　inteVPAS 1. 0 实现的曲柄滑块机构

4　结论

将机构系统样机的分析模块开发为独立于中模型的组件, 不仅简化了分析模块的复杂性, 有利于模块化和具有很好的可扩展性。本文采用的是基于组件技术的三维引擎 ACIS, 另外, 也可以应用 OpenGL 等。应用的 ACIS 几何造型、视图操作等功能组件开发可视化虚拟样机分析系统原型, 提高了系统的开发效率和系统的稳定性等。这种机械系统虚拟样机的设计开发为复杂机械系统虚拟样机的进一步研究开发提供了平台。

参考文献:

[1]　熊光楞, 李伯虎, 柴旭东 . 虚拟样机技术[J] . 系统仿真学报, 2001, 13(1) : 114-118.

[2]　汪盛, 袁捷, 李宗岩 . 基于组件技术的模型管理[J] . 计算机工程, 2001, 27(1) : 38-40.

[3]　赵雯, 等 . 武器系统虚拟样机技术研究[J] . 国防科技大学学报, 1999, 21(1) : 58-61.

[4]　刘贤喜, 刘竹青, 周一鸣 . 机械系统虚拟样机软件原型化研究[J] . 中国农业大学学报, 2002, 7(2) : 76-80.

作者简介:

黎国华(1964-) , 男, 副教授, 博士生, 研究方向为 CAE/工业设计; 熊会元(1973-) , 男, 硕士研究生, 研究方向为 CAD/CAE。